Biologically Inspired Guidance for Autonomous Systems

Thesis submitted in accordance with the requirements
of the University of Liverpool for the degree of
Master of Philosophy

By

Luke Kenneth Topham

September 2019

ABSTRACT

Animals and humans can perform purposeful actions using only their senses. Birds can perch on branches; bats use echolocation to hunt prey and humans are able to control vehicles. It must therefore be possible for autonomous systems to replicate this autonomous behaviour if an understanding of how animals and humans perceive their environment and guide their movements is obtained.

Tau theory offers a potential explanation as to how this is achieved in nature. Tau theory posits, that in combination with the so-called 'motion guides', animals and humans perform useful movements by closing action-gaps, i.e. gaps between the current state and a desired state. The theory suggests that the variable τ, the time-to-close an action-gap at the instantaneous rate of closure, is manipulated to perform such closures.

This thesis therefore aims to explore how the variable τ cam be used for the control of autonomous vehicles. To this end, a simulation model of a Dr Robot Jaguar 4x4 rover has been created in the Gazebo simulator. A simulation environment was developed in which the robot was placed, the environment contained virtual sensors for data collection and obstacles for the use in experiments. This simulation environment replicated the hardware and software used later in the hardware experiments which contained a Dr Robot Jaguar 4x4 and an OptiTrack motion capture system.

A τ estimation function was developed for the system based on the rate of expansion of an obstacle in the images recorded by the robot's monocular camera. The results show that the estimation of τ was offset, with the offset becoming larger as the time-step between frames became larger. An equation has been formulated which estimates this offset given a known time-step allowing for it to be reduced.

Research has shown that drivers maintain a constant rate of change of τ ($\dot{\tau}$) when braking to a stop. A $\dot{\tau}$ value of 0.5 was found to be the average maintained by drivers

when stopping, this value ensures that the vehicle safely stops exactly on the desired stopping point if maintained correctly. Using the aforementioned τ estimation system, this manoeuvre was performed in the simulation environment and later replicated in hardware. In this experiment, the robot approached an obstacle at a constant velocity, whilst continuously estimating τ between itself and the obstacle. When the τ estimation became lower than a pre-defined safe threshold, a braking manoeuvre was initiated. The manoeuvre aimed to replicate the human drivers by controlling deceleration to ensure that a $\dot{\tau}$ of 0.5 was maintained until the robot safely stopped. The constant $\dot{\tau}$ approach to braking was shown to generate a safe stopping manoeuvre in both simulation and hardware, replicating that of human drivers.

Closing multiple gaps simultaneously can produce more complex manoeuvres. An experiment has been conducted whereby, given a known gap between a robot and an obstacle a path around the obstacle is generated and followed. The path is generated using two existing intrinsic τ guides, to control the closure of the relevant action-gaps. The robot follows this path using the pure pursuit algorithm, a path tracking algorithm that computes linear and angular velocity necessary to move the robot from its current position to reach some look-ahead point on a path. The path generating and following system was shown to enable the robot to safely avoid an obstacle. This experiment suggests a future opportunity to control such manoeuvres via the direct estimation of τ.

The results provided in this thesis allows the following conclusion to be drawn. Firstly, an accurate estimation of τ can be derived in real-time using only a monocular camera. Secondly, safe autonomous braking can be achieved by maintaining a constant $\dot{\tau}$ value of 0.5. Finally, Tau-guides can be harnessed to generate safe paths to avoid obstacles.

ACKNOWLEDGEMENTS

‘Intelligence is not a privilege, it's a gift. And you use it for the good of mankind.’

Dr Otto Octavius

This thesis would not have been possible without the kind support of a number of people. I would like to extend my sincere gratitude to all of them.

This project would not have been possible without the support and guidance of my supervisors Dr Mike Jump and Dr Paolo Paoletti. I am extremely grateful and fortunate to have received such dedicated and valuable guidance.

I would like to thank my wife Lisa for her continuous support and encouragement throughout this project.

The final mentions must go to my mum Lindsay and my dad Ken for supporting me through life.

ACRONYMS

2D	Two Dimensional
FOE	Focus of Expansion
FPS	Frames Per Second
LIDAR	Light Detection and Ranging
PWM	Pulse Width Modulation
ROS	Robotics Operating System
SDF	Simulation Description Format
SURF	Speeded Up Robust Features
TTC	Time-To-Contact
UAS	Unmanned Aerial System

LIST OF FIGURES

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 The Future Development of Autonomous Systems

There is a growing number of exciting developments in the area of autonomous guidance and navigation in both ground and aerial unmanned vehicles [1]–[6]. However, the applicability of many of the current systems is often limited in real-world scenarios. For example, there is often a reliance on off-board processing to allow perception of the environment or to calculate trajectories [7]–[9].

Already, leading automotive companies such as Audi, BMW and Tesla have semi-autonomous vehicles operating on the road [10]. Features of these vehicles include automated parking, advance warnings, and limited autonomous driving. Such automotive companies and leading technical companies are making significant investments to bring forth an era of fully autonomous vehicles operating safely on public roads. There have been a number of high profile accidents involving semi-autonomous Tesla vehicles, one such accident was caused by a camera not being able to perceive a truck and another because the camera could not perceive a person on a bicycle [11]–[13]. Tesla recommends that drivers using the autonomous driving features should pay full attention to the road to ensure that are able to resume control in an emergency. A number of crashes involving autonomous vehicles have occurred when the driver has been distracted, for example by their mobile phones [11]–[13]. Tesla's Autopilot system has also encountered issues with stationary objects this is due to the fact that semiautonomous driving systems are often designed to ignore stationary objects [11]–[13]. Stationary objects are ignored, especially at high speeds as to prevent unnecessary braking which itself can be dangerous, instead the

responsibility of responding to static obstacles resides with the driver [14]. Ensuring safety is a key factor in deploying autonomous vehicles on a large scale and to ensure such technology is accepted by the public.

1.2 Motivation

As the population increases, so does the number of cars on public roads, in 2017 traffic levels had increased by 1.1% from the previous year [15]. In the UK alone there was a reported 24,841 serious injuries and 1,793 reported road deaths in 2017 [15]. Although improved road and car safety has meant that there has been 39% fewer reported road deaths since 2007, road death figures have remained almost unchanged since 2012 [15]. Although it would be tempting to envisage a world without cars, with the roads occupied by public transport and professional drivers only, this is unlikely to happen, at least not soon. Although many countries have significant public transport infrastructure, many people enjoy the personal freedom that comes with owning their own car, travelling where and when they want.

There is no singular cause of road traffic accidents, however, human error is a common cause. Factors such as age, driver experience, medical conditions, alcohol and drug use and excessive speed are major reported causes of such accidents [16]. If perfected, fully autonomous vehicles can be expected to improve road safety, reduce the number of traffic accidents, and therefore reduce the number of casualties and loss or damage to property.

As well as the aforementioned safety benefits, traffic conditions can be improved with fleet management methods such as strict gaps between vehicles which could potentially reduce congestion issues [10]. Autonomous vehicles would offer mobility benefits to people who are unable or unwilling to drive, such as the elderly, disabled or intoxicated [10].

There are several both technical and non-technical issues that need to be resolved before the large-scale uptake of autonomous cars can take place. These issues apply

to not just autonomous vehicles on the road, but also autonomous robots, for example in factories, that may operate in the same environment as humans in public or in the workspace. As the adoption of robotics increases, it is likely that robotics will be encountered in even more environments.

Technical issues include reliability, real-time processing situational awareness, software complexity, validation, and testing. In particular, real-time processing including obstacle detection and decision making is of paramount importance to allow the vehicle or robot to safely manoeuvre in dynamic environments [10], [17], [18]. One such problem is the fact that, although cameras are useful as a sensor for capturing data about the local environment, the vast volume and speed at which data processing is required means that their use is too computationally intensive for on-board autonomous systems [10].

Another issue which may hamper the adoption of the technology is the cost, a Lidar system costs approximately £1000, this is a significant addition to the cost of a robot or vehicle, whereas the cost of a camera can be as little as £10 [1]. Cost can be prohibitive in the adoption of any technology.

Non-technical issues include ethics, cost and consumer trust, and these are likely to be a problem until the technology has reached maturity [19]. Ethical issues such as liability must also be tackled, traditionally it is the driver of the vehicle that is at fault, but this is a more difficult issue if the "driver" is not human. Volvo has voluntarily agreed to accept full liability for its own autonomous vehicles, but it is yet to be seen whether other automotive companies will also voluntarily accept liability for their own vehicles [20]. There are some ethical issues which have not been clarified, such as how would autonomous vehicles deal with problems similar to the trolley problem [21], [22]. The trolley problem is an ethical test with a number of scenarios, the subject is asked to imagine a trolley is about to hit a number of people, they are asked to see if they would redirect away from the group but instead sacrifice a single person. Research shows that a driver may opt to turn their car in

order to hit one person rather than to remain on course and hit five, how autonomous systems should deal with problems such as this is an open source of debate amongst ethicists [21], [22].

Another critical issue for both autonomous vehicles and autonomous robots in general is that research shows that a robot's behaviour has an effect on people's perception and acceptance of them [23], [24]. People have a natural fear of change, autonomous vehicles that display human-like behaviour may be more accepted by human drivers, and a further benefit is that human drivers would be more likely to safely anticipate the behaviour of autonomous vehicles. Developing an autonomous car which mimics natural human-like behaviour could potentially aid the acceptance and integration on public roads alongside human drivers and also allow for such vehicles to safely tackle unforeseen events [10], [23], [24]. Human drivers are not singularly concerned about getting to their destination, they also have other considerations which affects their driving style, such as passenger comfort and anticipation of other driver's actions. For example, a driver who has seen stopped traffic some distance ahead will not wait until the last moment to brake at a point that they must apply full brakes. This would be uncomfortable for the passengers and may not give other drivers time to react. Instead, the driver will gently apply the brakes over a longer time and distance ensuring that they stop before colliding with the obstacle in a smoother approach.

1.3 Tau Theory

Tau Theory is an ecological theory of perception, ecological psychology tries to explain the relationship between an organism and its environment [25]. Tau theory is based on the link between the motion of an observer or objects within their field of view and what the observer perceives visually. Motion causes an optic-flow field which contains invariants in the perceived image which can be exploited to perceive motion [26]. This information can be utilised to enable the observer to safely navigate and survive their environment. For an animal to survive in the wild they

must be capable of avoiding obstacles, avoiding predators, and finding food. To enable predators to catch their prey they must have an understanding of when they will make contact with their prey [26]. This action requires some prediction of the future, specifically, where will the predator's and the prey's bodies be in the future [26]. There must be a time-based variable available to an organism to enable such actions which require such predictions of the future. This variable is the time to contact (TTC) at the current closure rate, otherwise known as Tau (τ). Tau theory has been proposed as a method to control autonomous systems so that their actions are close to that of humans and other animals.

Tau theory posits that animals including humans use the Tau variable (also known as time to contact") in addition to simple guidance strategies in order to perform purposeful manoeuvres [27]–[29]. Here, an action-gap is defined as the gap between a current state and a desired state [29] and TTC can be considered as the time to closure of this gap at the instantaneous rate of closure. Examples of such action-gaps include drivers braking, birds perching [30], animals attacking prey [31] and acceleration being reduced to zero to stop a vehicle. Tau is the measure of change in an action-gap, it is the time-to-closure of the action-gap at the current rate of closure [27], [32].

From the aforementioned examples, the single temporal variable that is proposed for controlling motion is the Tau of a motion-gap [26]. The Tau of gap X is also referred to as the Tau-function of X, τ_x as shown in Figure 1. For an observer under motion approaching a surface with velocity ẋ, if the current distance to a desired target state is x then the Tau (τ) of the motion gap, x, is given in equation 1 [26], [31]:

$$\tau_x = \frac{\mathrm{x}}{\dot{\mathrm{x}}} \qquad (1)$$

Equation 1, the equation for Tau/TTC, can be considered as the time to impact if nothing changes, for example the vehicle in Figure 1 continues at a constant velocity towards the box obstacle. It is this variable that advocates of Tau theory suggest that

humans and animals use to perform purposeful actions. This thesis explores the possibility of using a camera to detect this variable and to then use the variable to perform purposeful actions in robots, with the aim of recreating human-like behaviour such as that discussed in section 1.2.

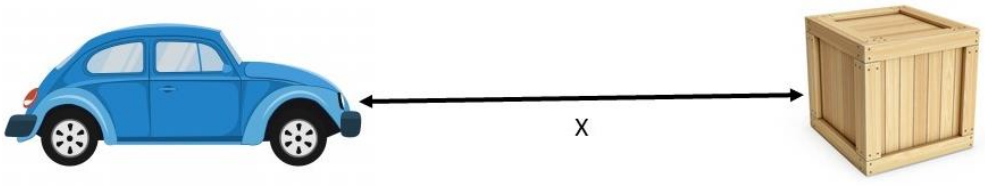


Figure 1. X Gap

According to Tau theory, humans and animals achieve purposeful actions by controlling closing the TTC value to zero using a range of Tau guides and other methods [31], [33]. For example, the driver of the vehicle in Figure 1 may use a constant $\dot{\tau}$ profile. The constant $\dot{\tau}$ profile results in a constant rate of change of τ whereby τ is reduced to zero at a constant rate of change. At the end of such a manoeuvre τ is zero and velocity is also zero if the appropriate coupling constant is used, this is discussed in Chapter 2 [34], [35]. Using this profile, the driver of the vehicle would decelerate to maintain a constant rate of change of τ $(\dot{\tau})$. The theory suggests that this is a subconscious process. This is a single example of the closure of one action-gap, there is a wide range of guides which can be applied to a plethora of gaps. Chapter 2 discusses the use of Tau guides and other methods of the closure of action-gaps.

1.4 Research Aim and Objectives

The aim of this research is to establish human-like perception and guidance methods for autonomous systems. The research presented in this thesis aims to address two of the issues discussed in section 1.2. Firstly, the issue of the acceptance of the behaviour of autonomous systems by adopting natural motion and perception strategies. Tau theory shows that the two can be combined to achieve purposeful

actions. Secondly, gaining valuable environmental awareness via cameras in real-time in order to aid autonomous decision making and actuation. The aim of establishing human-like perception and guidance methods for autonomous systems will be achieved by implementing a camera-based Tau estimation system and combining this with the use of appropriate Tau Guides which are discussed in Chapter 2.

The overall aim will be achieved by meeting the following objectives:

1. Develop a real-time vision-based approach to environmental awareness by detecting obstacles and estimating TTC between the observer and the obstacle.
2. Develop a natural, Tau-based process of guidance based on human driving behaviour for:
    a. A linear manoeuvre (i.e. stopping before an obstacle).
    b. A two-dimensional manoeuvre (i.e. steering around an obstacle).

Upon discovering an obstacle in their path, the driver of a vehicle has the choice of either stopping before colliding with the obstacle or driving around it. The decision of which to action to take is a complex one which involves many considerations such as safety, available space for the manoeuvre and the relevant driving laws. This is an important problem which must be solved in order to develop full autonomous vehicles. However, this problem is beyond the scope of this thesis as will be discussed in section 1.5.

1.5 Thesis Scope, Structure and Content

1.5.1 Scope

This thesis is concerned with natural guidance methods for autonomous systems. Therefore, this project will include two guidance experiments in both simulation and hardware using Tau theory, a natural method of perception and control. The first experiment will be a linear manoeuvre mimicking the problem of a vehicle stopping

at an obstacle. The second experiment will aim to guide the system around an obstacle so that it is able to continue along its path.

Other issues discussed in section 2.1 such as the many unsolved ethical issues surrounding autonomous systems are not included in the scope of this project. This research will not attempt to prove that the natural perception and guidance theories, which are used in this research, exist in nature. Proving Tau theory as the method of perception in nature is beyond the scope of this thesis. This thesis is instead concerned with how these theories can be used in engineering. Significant evidence has been provided by other research which suggests that Tau theory is present in nature, the most significant of which belong to David N. Lee [29]–[31], [36]. Researchers have also shown that Tau theory has engineering uses, some of which will be discussed in this chapter [7], [32], [34], [35], [37]–[41].

1.5.2 Structure and content

This thesis is divided into six chapters. Chapter 2 reviews the current state of the art in autonomous guidance methods. It also reviews the Tau theory and the guidance methods which have been utilised in the research activity described in the rest of this thesis. Finally, the chapter reviews a range of perception and guidance techniques used by other researchers.

Chapter 3 will discuss the methodologies used to conduct this research. Firstly, it will discuss the implementation of the simulation experiment, including the development and validation of a simulation model of the robot that was used in the hardware experiments. The hardware experimental setup will also be described. Finally, the perception methods that were development will be described, and the results of the validation of this system will be presented.

Chapter 4 describes the linear experiment, both in a numerical simulation environment and the hardware experiments. This experiment aims to decelerate and stop the robot before it collides with an obstacle. Results of experiments with both

truth data (accurately recorded positional data recorded using the OptiTrack motion capture equipment [42]) and data acquired via the perception methods are presented.

Chapter 5 describes the 2D lateral experiment with results from both a software and a hardware experiment. This experiment aims to manoeuvre the robot around an obstacle so that it avoids a collision with the obstacle but is then able to carry on its original path.

Chapter 6 reports some final remarks and a discussion of the work held within this thesis. It also offers advice for potential areas and tasks for further work and research.

1.6 Original Contribution

The original contributions presented in this thesis are as follows:

1. The implementation of Tau-based estimation using monocular vision only. The estimation method is based on optical expansion has been created which provides an estimation of TTC whereby TTC is estimate based on the rate of expansion of the obstacle in the image as discussed in chapter 3. This implementation allows for a rapid and accurate estimation of TTC which can be used in the control of autonomous systems. This Tau-based differs from conventional motion and object tracking methods (such as those provided by OpenCV [43]) as it provides information which can be directly used to control motion without the need for further processing or calculations.
2. The identification of the offset between estimated TTC and the true TTC caused when estimating TTC using the optical expansion method with a non-zero time-step, as described in chapter 3. A novel mathematical correction for this issue is described in chapter 3 and applied in chapter 4. This correction allows for a more accurate estimation of TTC.
3. The development and validation of a simulation environment based in Gazebo which includes a model of the Dr Robot Jaguar 4x4 used in the practical experiments which lead to this thesis. The model was validated

using a motion capture technology to ensure that the model performed as closely as possible to that of the hardware robot when given the same commands. The simulation environment allows for the safe and rapid testing of autonomous control systems.

4. The implementation of a novel constant $\dot{\tau}$ method of autonomous braking for collision avoidance as discussed in chapter 4. This manoeuvre ensured that the robot stopped before colliding with an obstacle. The constant $\dot{\tau}$ method of breaking replicates the way in which human drivers decelerate to a stop when driving. It is hoped that autonomous control which replicated human-like driving characteristics would be more accepted by human drivers.
5. The implementation of a novel obstacle avoidance method using a combination of Tau-guides to generate the desired avoidance path, as discussed in chapter 5. This manoeuvre allowed the robot to safely avoid an obstacle and to carry on travelling on its original course. This automated path generation means that the system is not reliant on rigid pre-defined manoeuvres, unlike several current systems.

1.7 Conclusion

People are unlikely to give up the freedom that owning a car provides, therefore this chapter has discussed the need for a safe, reliable autonomous vehicle. It has also discussed the fact that such a system should behave in a human-like manner to ensure that it is adopted on a larger scale. Similarly, the cost and computational requirements should be kept as low as possible to ensure that more people are able to afford the technology to increase safety on the road. In this chapter Tau theory has been proposed as a potential solution to this problem. It has discussed how human and animal behaviour can be explained and predicted by using Tau theory. Therefore, it is suggested that a safe human-like system can be produced by harnessing Tau theory.

CHAPTER 2

LITERATURE REVIEW

The research presented in this thesis incorporates knowledge from several subject areas such as robotics and psychology. This chapter will review the relevant areas and will identify the research gaps that have motivated the research in this thesis.

2.1 Autonomous Robotics

It is important for autonomous robots to be able to sense their environment to enable them to safely operate in dynamic environments. Barry *et al*. [4], [44], [45] at MIT's CSAIL group have developed a lightweight fixed-wing Unmanned Aerial System (UAS) capable of flying through cluttered environments and avoiding obstacles at speeds as high as 30mph. This system is based on a stereo-vision algorithm which allows the vehicle to detect obstacles and build a full map of its environment in real-time. Traditional algorithms would require the searching of the images at different depth fields to detect obstacles, however, this is computationally expensive. Instead, the authors realised that when flying at high speeds the space within 10 metres of the vehicle did not change much as it takes the vehicle less than one second to fly through this distance. Instead, this solution detects objects at a single depth of 10 metres and using the vehicles odometer data, "pushes" these obstacles forwards in the image to create a "complete" knowledge of the obstacles within the area 10 metres ahead of the vehicle without actually perceiving them within this range.

This approach requires less computation and can therefore be processed on smart phone type processors which are on-board the lightweight vehicle. This system is not dependant on any off-board processing, unlike many of the typical approaches [7]–[9]. This method differs to other solutions which constantly detect and track

objects, often using sensors such as LIDAR [46]–[48]. Such methods are computationally expensive and require larger/heavier equipment. Such systems are often constrained to external devices for perception or processing or require larger vehicles to carry the equipment.

This system has some significant weaknesses, firstly, during take-off and sharp turns, the vehicle is blind to obstacles that are closer than the measurement depth (10 metres). Similarly, due to the single measurement depth, upon performing evasive manoeuvres or being moved, the vehicle is unaware of any obstacles that were potentially occluded. Therefore, the vehicle assumes that there are occluded obstacles and moves significantly away from its original path, thus reducing the efficiency of its overall path to its destination.

Michels *et al*. [49] developed a high speed obstacle avoidance system using monocular vision and reinforcement learning for a ground based robots. Supervised learning is used to estimate depths using the monocular camera. Reinforcement learning is applied within a simulator to learn a control policy in terms of its steering, here the depth estimation is correlated with a laser range finder. The image from the camera is divided up into vertical strips, with each strip labelled with a value depending on the depth of the closest obstacle in that strip. There is a control policy to determine how aggressively to steer and when to slow down. The use of the vertical strip method limits the range of turning angles that the robot may use. This system was also shown not to be reliable with crashes recorded every 19 to 80 seconds depending on the terrain type, making long term navigation difficult.

Mitsch *et al*. [50] have developed a "provably safe" obstacle avoidance system for ground based robots. This system has both a "passive safety" and a "stronger safety" mode. The passive safety mode simply avoids stationary obstacles, the stronger safety mode leaves enough space between the robot and the obstacle to allow enough time for the robot to respond if the obstacle moves. Braking is done less harshly to give any following vehicle time to respond. The stronger passive safety mode and

overly conservative safety boundaries may cause issues in dense environments such as a much-reduced speed. This is a gap which this project aims to address by developing a safe obstacle avoidance algorithm which is capable of moving at high speeds through dense environments.

The US army have published a number of papers in the area of autonomous navigation, in which the primary terrain sensor of the vehicle is a LIDAR [2], [6], [48], [51], [52]. This approach uses machine learning allowing the system to learn to avoid obstacles by observing human pilots. To operate in real time, a layered approach is taken to plan globally and react locally. This combines a slower path planner that is continuously re-planning the path to the goal based on the perceived environment with a faster collision avoidance algorithm that ensures that the vehicle stays safe.

The path planner used is based on an implementation of Laplace's equation that generates a potential function with a unique minimum at the goal. The advantage of this method is that it provides smooth paths that are equidistant from obstacles, rather than getting close to them. The parameters necessary for collision avoidance can be learned automatically by analysis of only one path generated by a pilot remote controlling the vehicle to avoid obstacles.

Demonstrations were performed using an autonomous JUH-60A RASCAL, with a pilot on-board only for emergencies and to verify landing selections. Landing sites were chosen by the autonomous system, but the landing was made by a human pilot. The system could avoid large stationary terrain such as large hills and mountains and smaller/thinner obstacles such as a mast or pole, no work on avoiding moving objects such as other aerial vehicles were discussed by the author. The autonomous flight behaviour proved to be so good that it could be flown much faster than expected, however, the 600m-range 3D-LZ Lidar forced speed limitations. Future work is planned to use a longer-range LIDAR so that vehicle speed can be increased.

Demonstrations have shown to provide smooth autonomous path planning, flight and landing selection. There are some minor issues with this research, currently a trade-off in the sensitivity of the avoidance of small objects such as the pole used in the experiment and the over-reaction to large objects such as the terrain. The authors detected some issues which they name "stair-stepping" issues. These issues were experienced when flying over a ridge or during a planned descent, the vehicle would repeatedly speed up and slow down, this was due to the system attempting to keep air speeds within strict limits while following the topology of the voxel array [6], a grid in a three dimensional space. This issue could potentially be solved using a Tau-based solution. Planned future work includes threat, currently these are manually entered as no-fly areas. One such example of this was that the experimentalists had a requirement that they should not fly near cattle on a neighbouring farm to prevent upsetting the farmer, therefore, it was decided to mark the farm manually as a no-fly area.

Ren *et al*. [53], [54] have developed a fuzzy intelligent system which combines path planning and obstacle avoidance for ground based vehicles. This system includes two fuzzy logic controllers, one to generate a path when there aren't any obstacles and a second to generate commands to avoid an obstacle; an intelligent coordinator is then used to coordinate the actions of the two fuzzy controllers. The use of a fuzzy logic system was found to react quickly to obstacles and was also found to be robust to sensor measuring errors. However, this solution is limited in its set of obstacle avoidance controls, including speed and turning and the path planning algorithm does not give a smooth path. It appears that the system does not provide a smooth path as the robot changes directions multiple times to avoid a single obstacle instead of making a single reorientation to avoid the obstacle.

2.2 Tau Theory

Tau theory is central to this thesis. This section will firstly provide further background into this theory. Secondly, existing Tau-based solutions will be discussed.

2.2.1 Tau Theory and Tau Guides

Animals and humans can navigate their environment using only their senses, such as sight and echolocation with relative success. It must therefore be possible to create an autonomous system also capable of navigating complex environments using only on-board sensors such as cameras and LIDAR. To implement such a system, an understanding of how animals and humans achieve this feat must be gained.

Tau theory is an ecological theory of perception. Tau is the measure of the time-to-contact (TTC) or time-to-closure of an action-gap at the current rate of closure [25], [27], [28], [30], [31], [35]. An action-gap is defined as the measurable closure between a current state and a desired state [31]. All purposeful actions require the closure of at least one action-gap and often requires the simultaneous closure of multiple gaps. Lee has expanded this research by showing the use of Tau in a wide range of examples of the closure of action-gaps such as in hummingbirds birds perching [30], drivers braking [27], [28], somersaulting and babies suckling [25].

Both the hummingbirds and the drivers were found to maintain a constant rate of change of Tau ($\dot{\tau}$) during the deceleration phase of their manoeuvres [27], [30], [34]. For the birds this meant that they were able to close the action-gap between their feet and the branch but ensured that their feet were still moving when contacting the branch to ensure that they were able to grip the branch. The driver however would want to reach their desired stopping point with zero velocity. This is achieved by altering the value of $\dot{\tau}$ to fit the needs of the manoeuvre. For example, the driver would ideally maintain a $\dot{\tau}$ value of 0.5 ensuring that as the action-gap closes to zero, so too does the velocity of the vehicle. The bird would maintain the value so that

$0.5 < \dot{\tau} < 1.0$ ensuring that the action-gap is closed with a non-zero velocity [25], [32].

Research has shown that the behaviour during the closure of action can be explained by the use of mental models of desired models known as Tau-guides [25]. Evidence from nature supports this statement and has also provided examples of such Tau guides τ_g follow specific forms such as constant deceleration, constant acceleration or general Tau guide as shown in Equations 2 to 5 [25], [26], [28], [29], [31], [35], [55]. Equation 6 shows how an action-gap, x, is Tau-coupled onto a Tau guide where τ_x is the Tau of the action-gap, τ_g is the Tau guide, T is the total time for the manoeuvre, t is the instantaneous time and K is the coupling constant[26].

Constant velocity Tau guide

$$\tau_g = (t - T) \tag{2}$$

Constant deceleration Tau guide

$$\tau_g = \frac{1}{2}(t - T) \tag{3}$$

Constant acceleration

$$\tau_g = \frac{1}{2}\left(t - \frac{T^2}{t}\right) \tag{4}$$

General Tau guide

$$\tau_g = \frac{t(T + t)}{T + 2t} \tag{5}$$

General Tau

$$\tau_x = K\tau_g \tag{6}$$

When implementing these Tau-guides there are some important variables that must first be decided upon. Firstly, T, this is the duration of the manoeuvre. Secondly, K, this is the coupling constant between motion and the Tau guide. These variables can be used to adapt the manoeuvres to fit their purpose.

It is posited that purposeful actions are achieved by animals by the internal generation of these intrinsic Tau-guides which provide a "desired-Tau" as shown in Figures 2 to 6. The animal would therefore accelerate or decelerate to ensure that their time-to-closure of an action-gap follows this desired pattern of closure in a subconscious manner [27]. Lee has provided evidence of such Tau-guide based manoeuvres, for example, a person moving food to their mouth with their eyes closed. In this example, the time-to-closure of the gap between the hand and mouth can be predicted by using a Tau-guide such as a general Tau-guide. The general Tau-guide as shown in Figure 4 provides an acceleration phase followed by a deceleration phase allowing for a stationary hand to accelerate towards the mouth and then decelerate as it gets closer to the mouth, thus the person avoids hitting themselves in the face. This is achieved by generating the Tau-guide mentally and ensuring that the actual Tau follows the desired Tau generated by the general Tau-guide.

Figure 2 shows the gap distance, closure rate (velocity) and acceleration when following a constant acceleration guide. Figure 2 shows that when using a constant acceleration guide there is a strong initial acceleration which gradually eases off over time, changing the value of K changes the point of peak acceleration and also the rate of acceleration [25]. When K is used assigned a value of one, constant acceleration is achieved. When time T is reached the gap, closure rate and acceleration have returned to zero. Figure 2 also shows the effect of K, as the value of K is increased maximum velocity is reached further into the manoeuvre. A K value of 0.5 will mean that there is a finite deceleration at the end of the manoeuvre which will allow for the animal or robot to avoid a collision. With a K value > 0.5 an infinitely long deceleration is required at the end of the manoeuvre, which therefore results in a collision [25]. Figure 2 shows the resulting time-to-contact over time, this figure suggests that the higher the K value, the more aggressive the manoeuvre. Although it would be possible to change the K value during a manoeuvre, it would require the recalculation of the manoeuvre from the current

point. It would therefore be desirable to decide on the K value before the manoeuvre based on the objective of the manoeuvre to ensure a smooth manoeuvre.

In section 2.1 it was noted that some current systems suffer from not being able to calculate smooth paths to follow. Tau guides could be used to generate smooth paths and can be easily adapted to suit the needs of the manoeuvre.

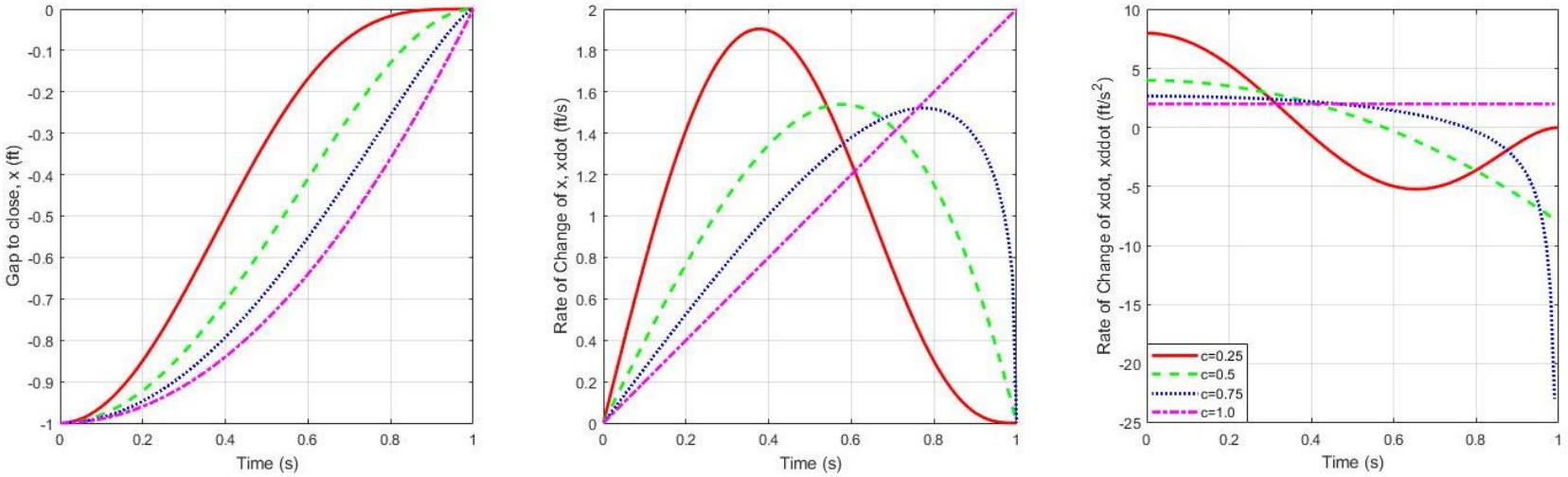


Figure 2. Gap distance, closure rate and acceleration when following a constant acceleration guide

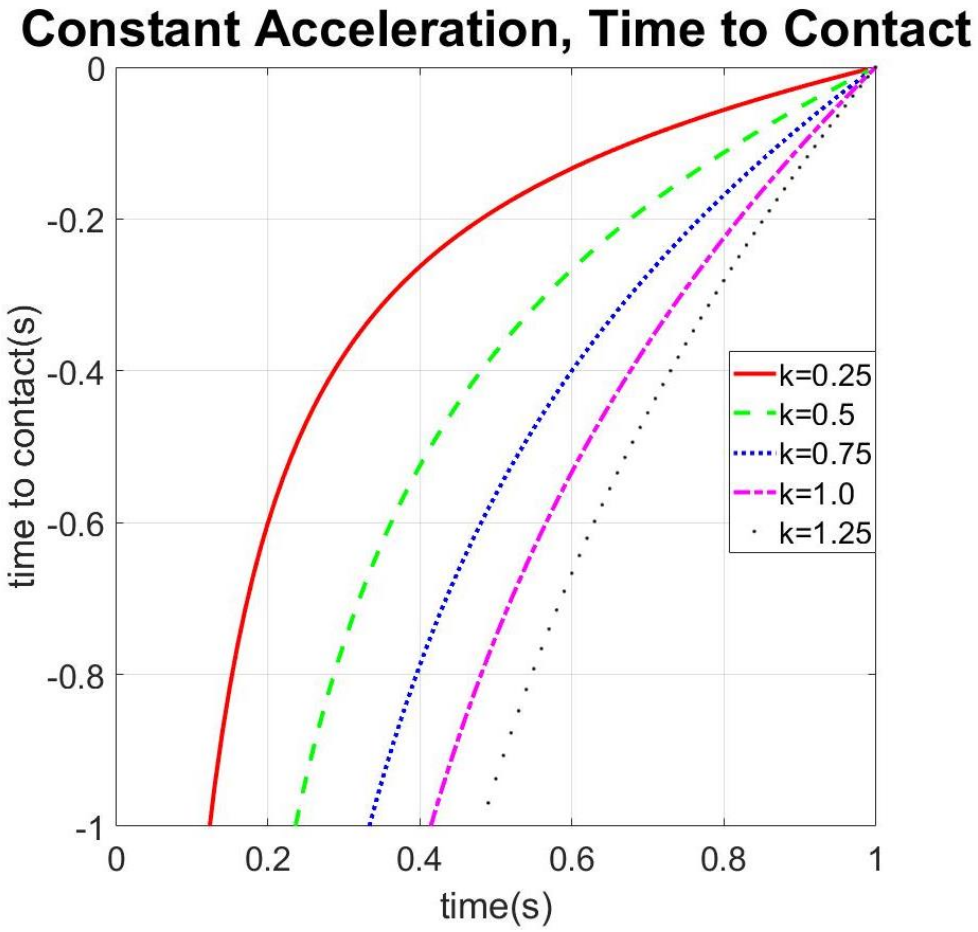


Figure 3. Tau when maintaining a constant acceleration guide

Figure 3 shows the TTC profile when maintaining a constant acceleration guide. Figure 4 shows the gap distance, closure rate and Tau when following a general Tau guide. Unlike the constant acceleration and constant deceleration guides, the general Tau guide guides an animal or system from a stop to a desired state. Therefore, this guide results in an acceleration and deceleration phase as shown in Figure 4 [26]. For example, a general Tau guide could be used by an animal flying from one branch to another, the animal accelerates away from the starting branch and decelerates as it approaches its target branch.

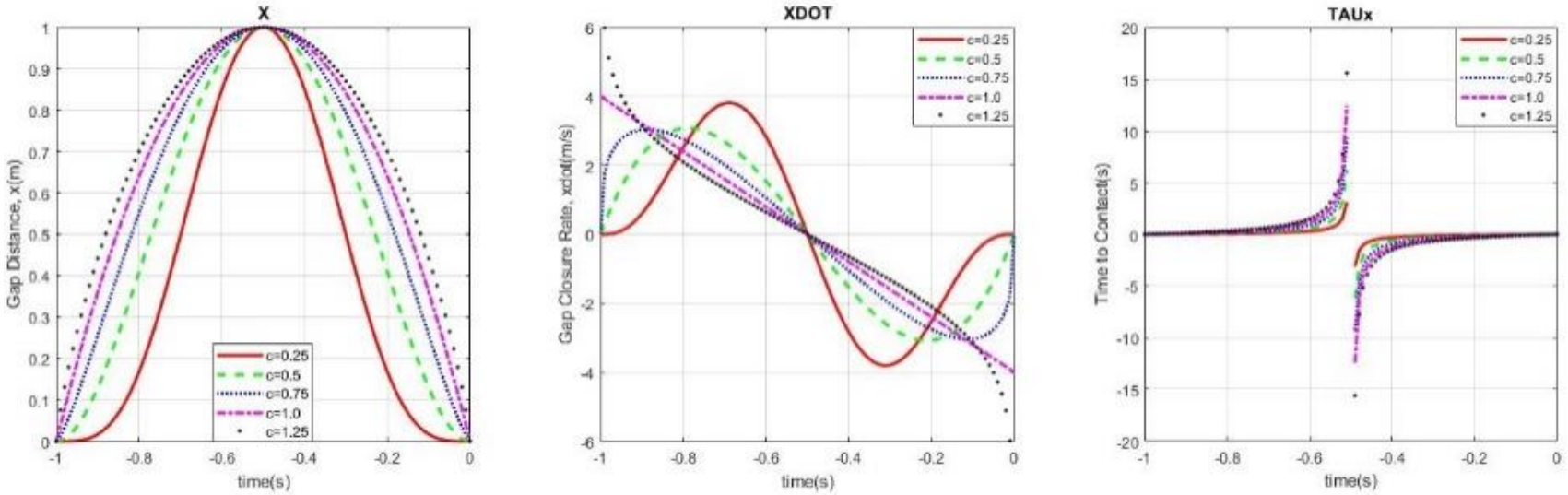


Figure 4. Gap distance, closure rate and Tau when following a general Tau guide

2.2.2 Tau-dot and the control of deceleration

An early hypothesis of how drivers decelerated their cars to a stop using Tau did not include the use of Tau guides [31], [34]. Instead it was posited that drivers are able to initiate an effective deceleration strategy by maintaining a constant rate of change of Tau ($\dot{\tau}$), as shown in Equation 7 [26], [27], [31], [32], [34]. In research both animals and humans were shown to maintain a constant $\dot{\tau}$ to control deceleration to a stop [31].

Figure 5 shows the gap distance, closure rate and rate of change of closure (acceleration) when maintaining a constant $\dot{\tau}$. The constant rate of change of Tau can be clearly seen in Figure 6, again the time-to-contact at time zero is zero.

$$\frac{\Delta\tau}{\Delta t} = \dot{\tau} = c \tag{7}$$

If K is 0.5 then the animal or system will stop exactly as it reaches the object. If K < 1 then the manoeuvre will end with zero velocity, however if 0.5 < K < 1 then there will be some contact with the object. If k < 0.5 then the object will never be reached, assuring that there will not be any collision with the object [26], [31]. If K = 1 then a constant velocity will be maintained, K > 1 will result in acceleration.

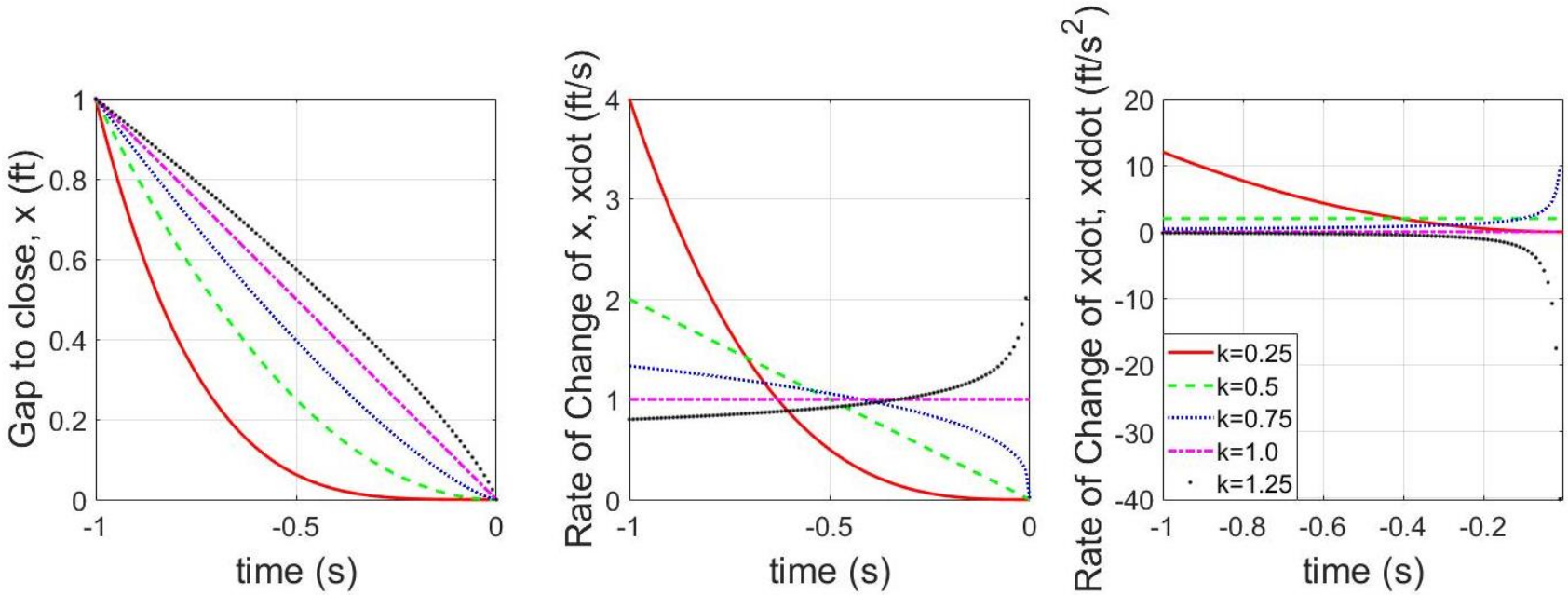


Figure 5. Gap distance, closure rate and acceleration when maintaining a constant Tau-dot

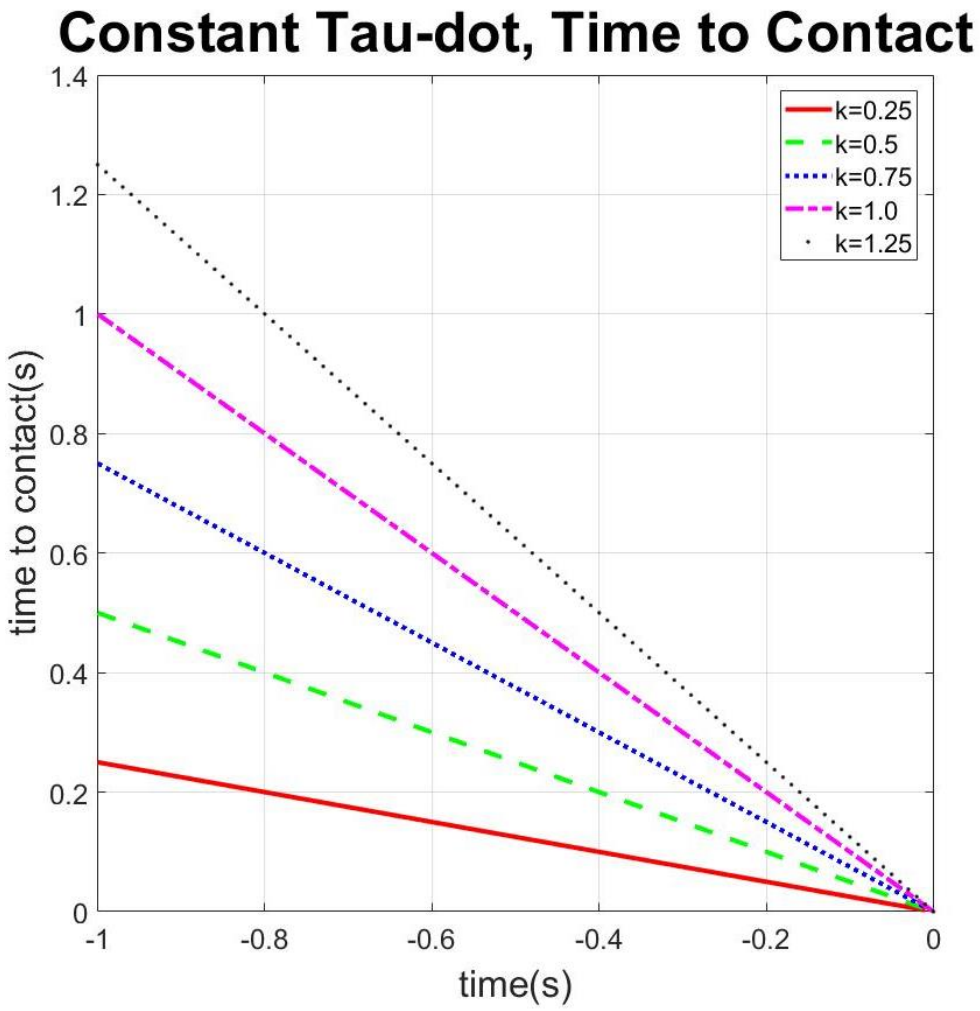


Figure 6. Tau when maintaining a constant Tau-dot2.2.3 Existing Tau-Based Solutions

There are many Tau-based approaches to the collision avoidance and autonomous navigation problem [7], [8], [60]–[69], [34], [70], [71], [37], [39], [40], [56]–[59], however, most of this work has focused on using Tau as a trigger rather than a method of control. For example, a number of these systems measure Tau and when

it is estimated as being lower than a safe threshold, they trigger a predetermined manoeuvre to either stop or change path.

An example of the aforementioned Tau triggered systems has been developed by Kaneta *et al*. [40]. The authors have developed a Tau-based collision avoidance system which allows a ground-based robot to avoid a collision with an oncoming robot by making a simple left / right turn. This system depends on obstacles being visibly marked to aid perception. This begs the question as to whether the system would function in a realistic environment. Souhila *et al*. [59] proposed a similar Tau-based collision avoidance system. This system was found to perform poorly in poor light conditions due to the nature of the object detection method which did not include the ability to recognise the same object in different levels of brightness.

Sanchez-Garcia *et al*. [37] have described a ground based system capable of avoiding obstacles without constantly calculating the optical flow field. However, this system simply calculates the optical flow field when it is known that the robot is close to a collision, then the optical flow divergence is used to decide which direction the robot should steer in. This approach is not feasible in a dynamic environment or in a fast-moving robot or vehicle.

2.3 Tau Estimation

It is possible for a system to use Equation 1 inside a controller using known values of X and $\dot{X}$ using external motion tracking or on-board sensors such as LIDAR. However, this is unlikely to be the case in nature. If for example the driver of a car was to perform these mental calculations the short delay in calculating the current Tau would likely result in a collision or undesirable actions [25]. Therefore, there must be another way in which animals and humans use their senses to obtain a usable value of Tau. Ecological psychologists would argue that Tau is directly perceived by animals and humans. This section aims to explore examples of how this may be achieved in nature and also by using computer vision techniques.

2.3.1 Optical Expansion

In his early work, Lee suggested that drivers may use the optical expansion of the vehicles in front of them to estimate the TTC of themselves with the vehicle ahead of them [26], [35], [36]. For example, Lee suggested that a driver wanting to stop behind a stopped car could use the expansion of the cars brake lights to derive an estimation of Tau.

Figure 7 shows how a single object in an image expands over time when being approached at a constant velocity. Equation 12 shows the equation used to estimate Tau from optical expansion and equations 8 through 12 shows how this equation is derived where *b* is the base of the object in the image and *h* is the height. Equation 8 is the formula for 2D area. Using the chain rule of differentiation gives Equation 9. Equation 10 is Equation 9 divided by Equation 8. In equation 11 it can be assumed that τ_b and τ_h are equivalent. Finally, rearranging for τ gives Equation 12.

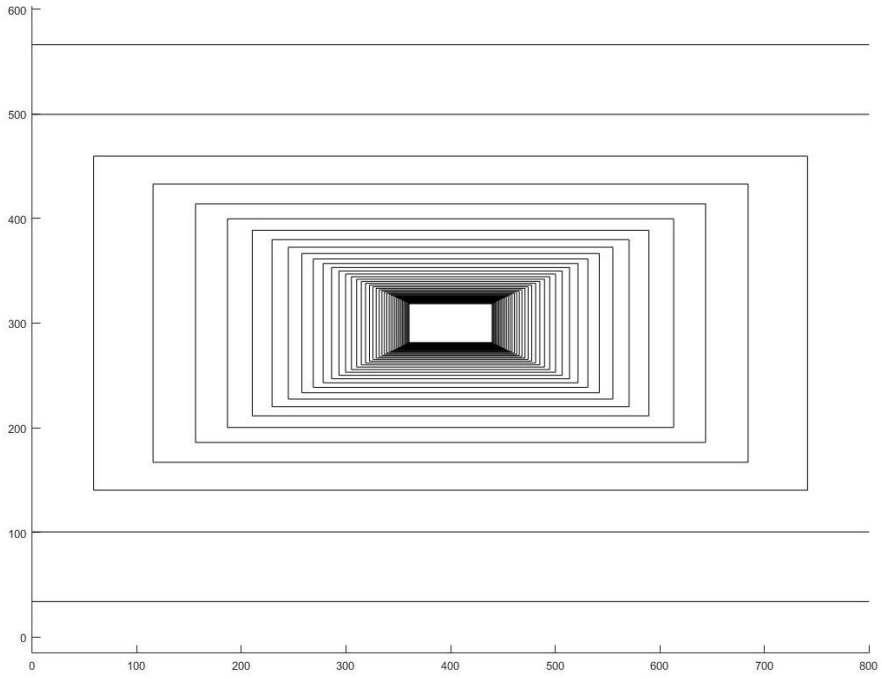


Figure 7. Optical Expansion

$$A = bh \tag{8}$$

$$\dot{A} = \dot{b}h + b\dot{h} \tag{9}$$

$$=> \frac{\dot{A}}{A} = \frac{\dot{b}h + b\dot{h}}{bh} \tag{10}$$

$$=> \frac{\dot{A}}{A} = \frac{\dot{b}}{b} + \frac{\dot{h}}{h} = \frac{1}{\tau_b} + \frac{1}{\tau_h} = 2\frac{1}{\tau} \tag{11}$$

$$\tau = \frac{2A}{\dot{A}} \tag{12}$$

Mori and Scherer used the optical expansion method in their UAS solution [8]. In this solution they use Speeded-Up Robust Features SURF, this is a method of detecting features in an image. At each frame of the video, they then detect which features of the image have grown in size and then they finally estimate Tau using this expansion over time. Mori and Scherer only used Tau as a trigger for a pre-defined obstacle avoidance manoeuvre, they did not use features of Tau theory such as Tau guides for the avoidance manoeuvre.

2.3.2 Optical Flow

"Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image" [72]. Optical flow occurs due to the relative motion of an object and observer [72], [73].

Based on the equations of the expansion of the optical flow field it is possible to estimate Time-to-Contact [74]. A common method of estimating Time-to-Contact is to estimate a focus of expansion (FOE) and to then compute Time-to-Contact via the pixel velocity and FOE [68], [74]–[76].

Although research such as that of Camus [74] has shown promising results for the optical flow method of estimating Time-to-Contact, in practice it has proven impractical for real-time solutions due to the computational expense, particularly for on-board systems [68]. Solutions based on this method are often computationally

expensive, thus limiting its use in autonomous robotics [74]. A common assumption of this method is that brightness is constant, which states that the brightness of a point in an image does not change significantly as it moves across the image[73]. This limits its applicability in uncontrolled environments as the constant brightness assumption can be violated in a number of ways for example on specular surfaces (surfaces which appear to have different brightness when viewed from different angles) or when a light source is moved [73], [77].

When compared to the optical expansion method discussed in section 2.3.1 it could be argued that the optical flow method would work in a larger variety of situations as there is no need for the detection of objects in an image. This is because the optic flow method is concerned with deriving a time-to-contact value based on the flow of pixels in an image, rather than relating directly to an entire object in an image. However, the optic flow method is much more computationally expensive than the optical expansion method which would therefore limit the applicability for small autonomous vehicles. The constant brightness assumption of the optic flow method does not affect the optical expansion method, possibly increasing the number of situations.

2.4 Alternative Theories of Perception

There are several theories of perception. The two main branches of perception are the Constructionist view and the ecological viewpoint [78].

The constructionist view suggests that the observer does not passively receive information regarding their environment, instead they build up knowledge of the environment using previous knowledge and fragments of sensations. The argument for the constructionist view stems from the fact that a two-dimensional image is formed on the retina of the eye, but humans see in three dimensions. Constructionists therefore argue that somewhere in the brain a three-dimensional reality is constructed from this two-dimensional image, though constructionists do not agree upon how this achieved. There is a suggestion that to make sense of an image, there must be

sufficient knowledge or previous experience in order to be able to interpret the image [79].

In contract to the constructionist view, the ecological viewpoint suggests that the observer does not require any previous knowledge on the environment, nor do they have to construct reality [27], [32]. Instead the ecological viewpoint states that the environment provides enough information to provide the observer the ability to perceive their environment.

Tau theory is an ecological theory of perception and therefore by using this method camera-based systems can be developed which can perceive their environment using only the image from the camera. Tau theory was chosen as the method of perception for this thesis since it enables the system to detect when the gap between the camera and an obstacle is closing. Tau theory also allows for a measurement of the time-to-collision (if nothing changes). These two points provide enough information to allow the observer to perform manoeuvres such as braking or avoiding an obstacle.

2.5 Discussion

The CSAIL system is a cutting-edge collision avoidance system. Compared to the other methods it is computationally less expensive thanks to the pushbroom method of perception [4], [45]. This computational advantage allows the system to perform all calculations on-board, unlike a number of the systems reviewed in this section which rely upon off-board computation or external sensors [8], [9], [80]. Performing all computation on-board allows the system to be used in a wider range of situations.

Although the CSAIL system can be considered the cutting-edge in its category, it is not without its limitations, which a Tau-based system could potentially improve upon. For example, the CSAIL system is not capable of detecting objects within 10 metres after a sharp turn, thus preventing occluded objects from being detected. This project aims to estimate Tau continuously, thus enabling the detection of occluded obstacles. When multiple obstacles are spread before the UAS, occasionally the

system will become confused and will collide with an obstacle. This project aims to address this problem by constantly calculating Tau and performing manoeuvres to avoid the closest obstacles. This system is also dependent on a limited library of manoeuvres. As such, no attempt is made to make an efficient turn whilst remaining on course to a destination. Therefore, there is an opportunity to implement a Tau-based guidance system to improve upon these limitations.

As can be seen from equations 1 through 6, the maths required for Tau theory is relatively simple, especially compared to methods such as deep learning and other popular Artificial intelligence methods [81]. This is an important point, if a solution is too computationally complex then it will not be possible to get the code to run on small single board computers such as the Raspberry Pi, Arduino Uno and ODROID which are commonly used to implement small autonomous vehicles. If autonomous software can be uploaded to these single board computers then it is possible to upload these autonomous guidance programs to smaller robotics such as Unmanned Aerial Systems (UAS) and smaller ground robots such as the Jaguar Rover 4x4 which is used in this project [82].

Light Detection and Ranging (LIDAR) systems are commonly used in autonomous guidance systems [1], [6], [8], [17], [83], [84]. LIDAR systems allow for the accurate measurement of the distance between the LIDAR sensor and objects in the environment around it. LIDAR technology is useful; however, its cost, size and weight can prevent it from being used in smaller robotics systems. A Tau-based system could use a small, light digital camera to meet its requirement for perception, thus allowing its use in smaller robotics systems.

2.5 Conclusion

This chapter has introduced the theory underpinning this thesis. It has shown how Tau theory can be used to explain and predict the behaviour of humans and animals in a way which can be replicated in robotics. It has discussed a range of perception methods and has shown that Tau-based methods are computationally less expensive

than Artificial methods and do not require expensive and heavy LIDAR systems common in robotics, meaning that they can be used on small light-weight robotics systems. It also described the cutting-edge CSAIL system which is capable of impressive feats of autonomous guidance. It then explored several limitations of the CSAIL system which may be improved upon using a Tau-based method of perception.

CHAPTER 3

METHODOLOGY

This Chapter discusses the methodology for both the hardware and simulation experiments conducted in the pursuit of this thesis. A Dr Robot Jaguar 4x4 robot was used for the hardware implementation of this project [82], and this platform is described in Section 3.2.1. Before proceeding with real hardware implementation, a simulation model was created in the Gazebo simulation environment [85]. Gazebo is an open-source 3D robotics simulator which can make use of multiple high-performance physics engines and can model sensors such as cameras [85]. This allowed for faster algorithm development and was used to try to help to ensure safe experiments with the hardware; algorithms were first prototyped in the simulation environment and were only moved onto the hardware once they were demonstrated to work as expected.

3.1 Simulation

Simulation supports the rapid development and validation of robotics in an environment that is safe from the dangers of hardware. Using a simulation environment allows for tests to be quickly set up and re-run many times with little break between experiments, unlike hardware experiments which may require repositioning and resetting of systems. This also allows for adjustments to be made and quickly tested, allowing the rapid tuning and testing of the software. The robot used in this experiment has a height of 25.5cm, a width of 53cm, a length of 57cm, weighs 33kg and can travel at 7km/hr, therefore this robot can cause serious injuries to personnel and significant damage to property, it is important to minimise these risks. The specification for the robot can be found in Table 3, this was the specification table that was used as a reference when designing the simulation model.

Section 3.1 will discuss the development and validation of the model and will also discuss how it is controlled via Simulink [86].

3.1.1 Gazebo

The Gazebo simulation environment has been utilised for the simulation experiments for this research [85]. Gazebo is open source software and provides useful APIs and other useful plugins such as drivers for skid steer robots and interfaces for using OpenCV, a useful computer vision toolkit [85], [87]. As the Dr Robot Jaguar 4x4 is a skid steer robot, the Gazebo skid steer driver was used to implement the control of the model. Other sensors such as odometry systems were included in the model but were not used for the experiments in this system. The additional sensors were included in the model to allow future researchers to use the most realistic and complete model of the rover, however, as the nature of this thesis focuses on visual perception some of the sensors were not required for this research.

Gazebo is also compatible with ROS, the chosen framework for implementing and experiments on the physical rover. This means that most ROS code developed for the model, will work directly on the physical robot with very few amendments. This improves safety as anything that works in simulation should also work in hardware. ROS and Gazebo are quickly becoming an industry standard and Gazebo is now DARPA's preferred simulation environment for its virtual robotics challenge [88], [89].

As shown in figure 8, a model representing the Dr Robot Jaguar 4x4 robot has been developed in Gazebo. Functionality, dimensions and features have been included and closely match that of the physical robot. Some internal systems such as GPS receivers were not included in the model as they were not necessary for the experiments. The wheels in the model are controlled via a skid steer driver to emulate the behaviour of the rover. This driver translates velocity commands into commands for pairs of wheels (right side and left side), providing a manoeuvrability that closely matches that of the physical rover.

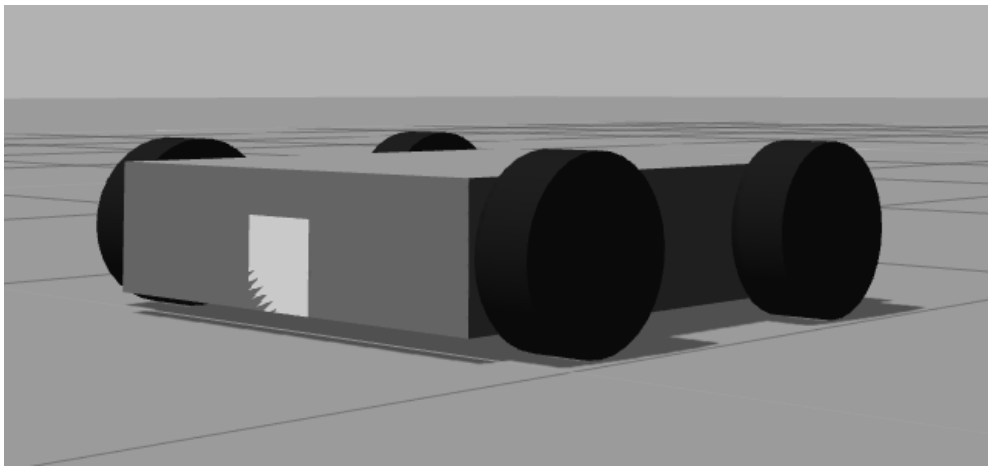


Figure 8. Gazebo Model

The model was created for the Gazebo simulation environment by creating a Simulation Description Format (SDF) file [90]. The SDF file allows the specification of all aspects of the model, including dimensions, weight and joints [85], [90], [91]. The SDF file also allows for the specification of some functionality, such as the camera functionality and the skid steering driver.

The modular nature of SDF files means that producing scenarios with multiple vehicles and more complex environments should be a relatively trivial task in the future [90], [91]. Gazebo includes models for many different types of obstacles which can be easily inserted to create complex and cluttered environments [85].

3.1.2 Simulink

Simulink is used to control the simulation model in Gazebo. Simulink is a MATLAB-based graphical programming environment [86], [92] which uses a graphical block diagram tool for rapid prototyping and testing. In the simulation experiments Simulink is used for all of the required processing. Simulink is responsible for the TTC estimation as discussed in Section 3.4, the proportional controller and for issuing commands for the model using a ROS message at each iteration to update the wheel commands. A proportional controller reads sensor data

and computes the desired actuator output based on the error between the sensor data and the desired sensor data [93].

3.1.3 Robotics Operating System

The Robotics Operating System (ROS) [94]–[96] was utilised for both the simulation and hardware implementations. ROS is a flexible framework for writing robotics software. It consists of tools, libraries, and conventions to simplify the task of writing robot applications. The Jaguar Rover 4x4 can be controlled using ROS. One of the main benefits of using ROS in this project is that it is compatible with both the physical robot and the simulation environment, described in the Section 3.1.1. The benefit of using ROS is that any code written for the simulation model will work in the same way for the hardware implementation with minimal changes.

ROS also provides a powerful robot description language which has been used in this research to create the model of the rover in a user-friendly way, the robot description format uses XML files to define models. This definition is compatible with all of the other ROS modules and components.

3.1.4 OptiTrack

OptiTrack is a motion tracking system which can stream tracking data directly to MATLAB in real time [42]. It is used in this project for several reasons. Firstly, it has been used in the model validation as discussed in Section 3.2 to compare manoeuvres between the model and the hardware systems. Secondly it is used as part of the lateral experiment discussed in chapter 5. All positional measurements are taken from the centre of the robot.

3.2 Model Validation

As the simulation model is to be used for prototyping and testing, it is important that the behaviour of the model matches the hardware response as closely as possible. This will allow for any systems developed in simulation to be easily migrated to hardware. This subsection will discuss the validation of the Gazebo model which has

ensured that both systems are as similar as possible in terms of their control and performance as measured by their response to linear and lateral commands as discussed in this section.

To make the rover move, a command must be issued for each pair of wheels (left and right side). This command is issued as a Pulse-Width Modulation (PWM) signal with a value between -1000 and 1000, where 1000 is 100% of the power available to the motors driving the wheels. As one motor is installed upside down in relation to the opposite motor, to have the rover move forwards or in reverse then one of the PWM values must be negated (e.g. if the command for the left wheel is 200, then the command for the right wheel is -200).

The only significant difference between the robot and the model is that the model requires the translation of the velocity commands as the model requires linear and angular velocity commands whereas the robot requires velocity commands for both the left- and right-hand pairs of wheels. As the model takes its input as a desired linear (m/s) and angular velocity (rad/s) it is important that commands can be generated in a way that is equivalent between simulated and hardware robots. The first step in achieving this was to map the PWM commands to linear velocities (in m/s).

Figure 9 and Table 1 show such PWM velocity mapping. This data collected using the OptiTrack system [42]. In each case the PWM command was issued to the rover and the rover would then drive forwards for approximately 4 metres, this is the maximum distance allowed by the arena hosting the OptiTrack system. The velocity was then averaged over this distance to give the actual velocity. The periods where the rover was accelerating, or decelerating were excluded from the velocity estimation. The acceleration and deceleration periods that were removed last approximately 0.5 seconds and 0.2 seconds, respectively. The reason for their removal was to ensure that only the average speed for each command was being

measured and that this measurement was not affected by the starting and stopping of the robot.

Figure 9 shows an almost linear line, however, between 0 and 50 PWM there is a lack of linearity. The reason for this is that when the rover is given a very low PWM command (PWM < 30) it struggles to move which results in an inconsistent velocity. When given a higher PWM command (PWM > 30) this problem is not encountered, a consistent velocity is produced.

Low battery charge does not degrade the rover's performance. When the battery's power is too low, the rover will not move. This allows for consistent testing and measurements for the velocity mapping. Figures 16 and 20 refer to an experiment which involves a significant forwards movement, these figures show a consistent forwards movement in an acceptably straight line.

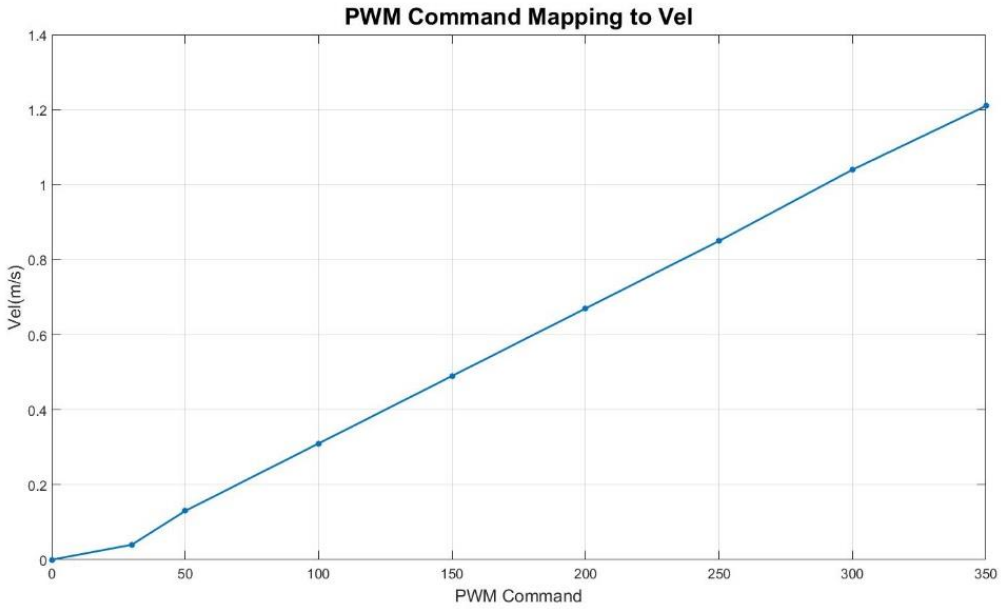


Figure 9. PWM-Velocity Mapping

PWM Command (Left)	PWM Command (Right)	Actual Velocity (m/s)
+30	-30	0.04
+50	-50	0.13
+100	-100	0.31
+150	-150	0.49
+200	-200	0.67
+250	-250	0.85
+300	-300	1.04
+350	-350	1.21

Table 1. PWM-Velocity Mapping

Equations 13 and 14 [97] are used to produce the commands for the left and right wheel pairs.

$$Left\ Command = \frac{\left(\frac{v + \omega * w_b}{2}\right)}{w_r} \quad (13)$$

$$Right\ Command = \frac{\left(\frac{v - \omega * w_b}{2}\right)}{w_r} \quad (14)$$

Where v is the linear velocity, ω is the angular velocity, W_b is the wheelbase of the robot and W_r is the wheel radius of the robot. The command conversion algorithms do not take in to account the PWM to velocity performance of the rover. Therefore, it is necessary to scale up the result of the conversion using the PWM to command

mapping table. To do this the output of the algorithm is divided by the known output of the algorithm for the input of 1m/s (~7) and then multiplied by the necessary value to achieve 1m/s (300 PWM).

Table 2 shows the tests which have been completed for the purposes of validation. The values for these tests were chosen to give a variety of velocities that could be performed safely within the bounds of the motion capture arena for at least five seconds. The PWM commands are the inputs for the rover and the velocity commands are the inputs for the model. Starting with the desired linear and angular velocities the PWM commands were generated using the aforementioned method.

For the turning manoeuvres it is important to ensure that the positional data of both the model and the rover can be overlaid with the same initial heading. To achieve this, the rover and model first drive a straight line, this allow for the angle between the two lines to be calculated and then for one of the plots to be rotated using the rotation matrix in Equation 16 where m in Equation 15 is the angle to rotate in degrees, [x, y] are the original coordinates of the robot and [x_r, y_r] are the rotated coordinates.

$$\Theta = m\left(\frac{PI}{180}\right) \tag{15}$$

$$\begin{bmatrix} x_r \\ x_y \end{bmatrix} = \begin{bmatrix} Cos\Theta & -Sin\Theta \\ Sin\Theta & Cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{16}$$

Test No.	Manoeuvre Description	Rover Command Left Wheel (PWM)	Rover Command Right Wheel (PWM)	Model Command (Linear(m/s) and Angular Velocity(r/s))
1.	Drive Forwards 5 seconds	+150	-150	Linear: 0.5 Angular: 0
2.	Drive Forwards 5 seconds	+225	-225	Linear: 0.75 Angular: 0
3.	Drive Forwards 2.5 seconds Right Turn 3 seconds	+150	-150	Linear: 0.5 Angular: 0
		+54	-246	Linear: 0.5 Angular: 0.25
4.	Drive Forwards 2.5 seconds Left Turn 3 seconds	+150	-150	Linear: 0.5 Angular: 0
		+246	-54	Linear: 0.5 Angular: -0.25

Table 2. Validation Test Plan

For each test, an equivalent command was executed on both the rover and the model for the same amount of time compensating only for the initial delay (~0.68s) of the first command to the rover. The delay is due to the need for the controller to register with the ROS node. The removal of this delay was not achieved; however, the delay was modelled in the simulation model to ensure comparability. This initial delay was

measured issuing commands to the rover and recording data for 5 seconds, the rover only moved for the final 4.32s of the experiments, therefore the delay is 0.68s. This delay only affects the first command to the rover, further delays are negligible. OptiTrack motion tracking data was recorded to track the rover and the Gazebo simulation software tracked the position of the model. These two sets of positional data are compared below.

The error at each data point in the linear manoeuvres is calculated as the current distance travelled by the model minus the current distance travelled by the rover. For the turning manoeuvres, the error (e) between the coordinates of the model (x_1, y_1) and the rover (x_2, y_2) is calculated using equation 17.

$$e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{17}$$

The results of the first validation test from Table 2 are shown in Figures 10, 11 and 12. Figure 10 shows the linear distance travelled over time of both the model and the robot. Figure 11 shows the linear velocity over time of both the model and the robot. Finally, Figure 12 shows the error in distance (metres) over time, calculated as the robot's distance minus the model's distance. The mean error in distance was 0.0149 metres, an error of less than 1% for a manoeuvre of 2 metres.

The initial rise and fall in error seen in the first 0.5 seconds of figure 12 was caused by the rover starting to move slightly after the model had already started (~0.2s). The rover then moves slightly faster than the model and therefore the error begins to increase. At approximately 4.5 seconds the error decreases rapidly, this is because the model was slightly behind where the rover was but continued to move forwards a small amount (~0.01m) after the rover had stopped, thus decreasing the distance between them. The maximum error in distance at any point was less than 0.03m over a distance of 2m, this is an acceptable error (error <1%).

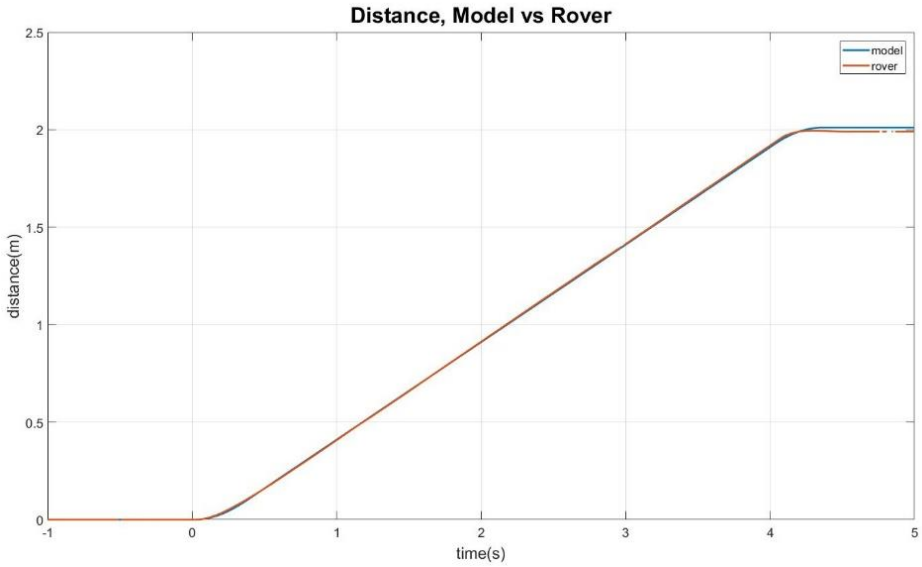


Figure 10. Linear Distance at 0.5m/s

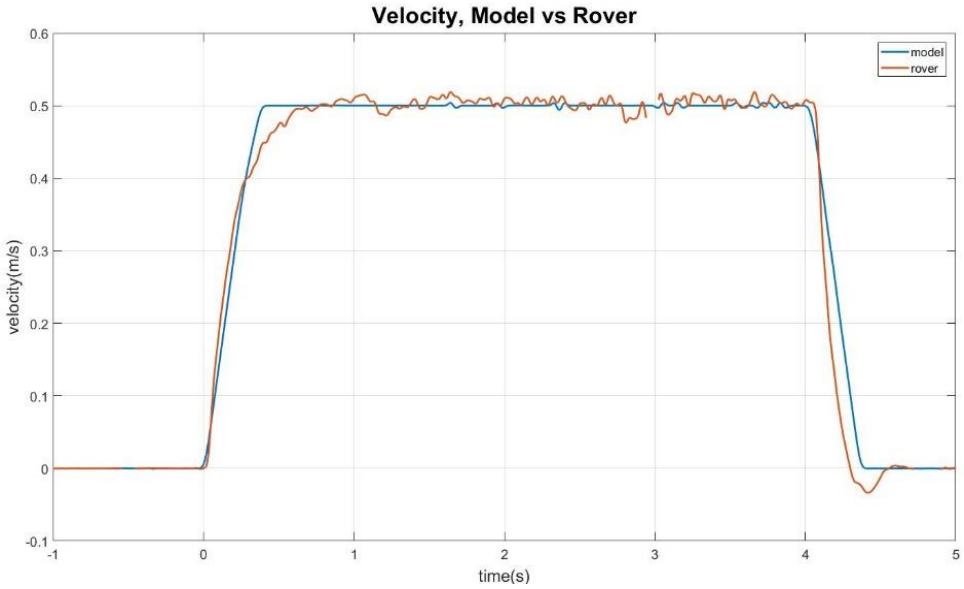


Figure 11. Linear Velocity at 0.5m/s

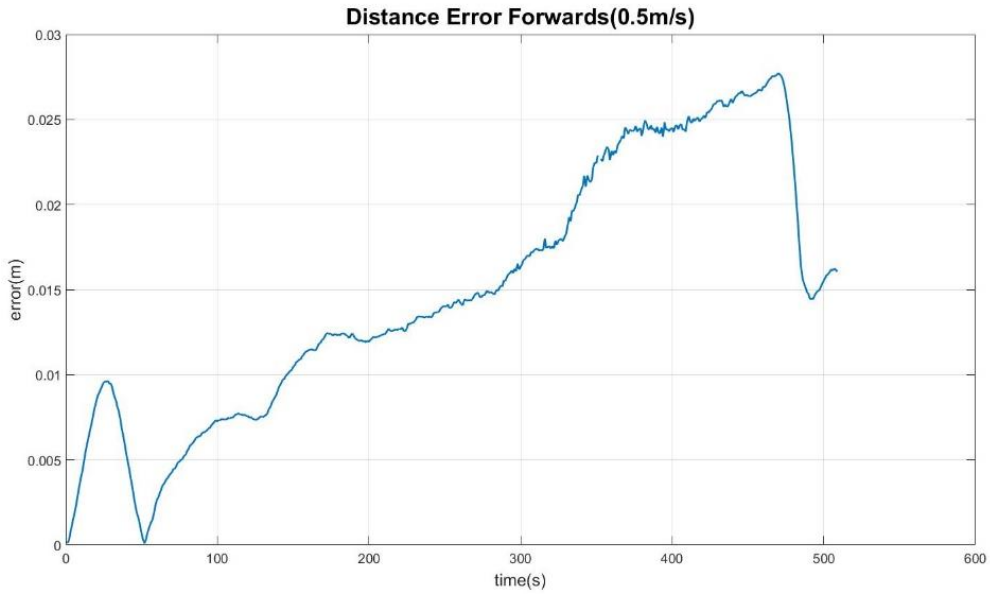


Figure 12.Error in Distance at 0.5m/s

The results of the second validation test from Table 2 are shown in figures 13, 14 and 15. Figure 13 shows the linear movement over time of both the model and the robot. Figure 14 shows the linear velocity over time of both the model and the robot. Finally, Figure 15 shows the error in distance (metres) over time, calculated as the robot's distance minus the model's distance. The mean error in distance was 0.0253 metres, again, this is an error of less than 1%.

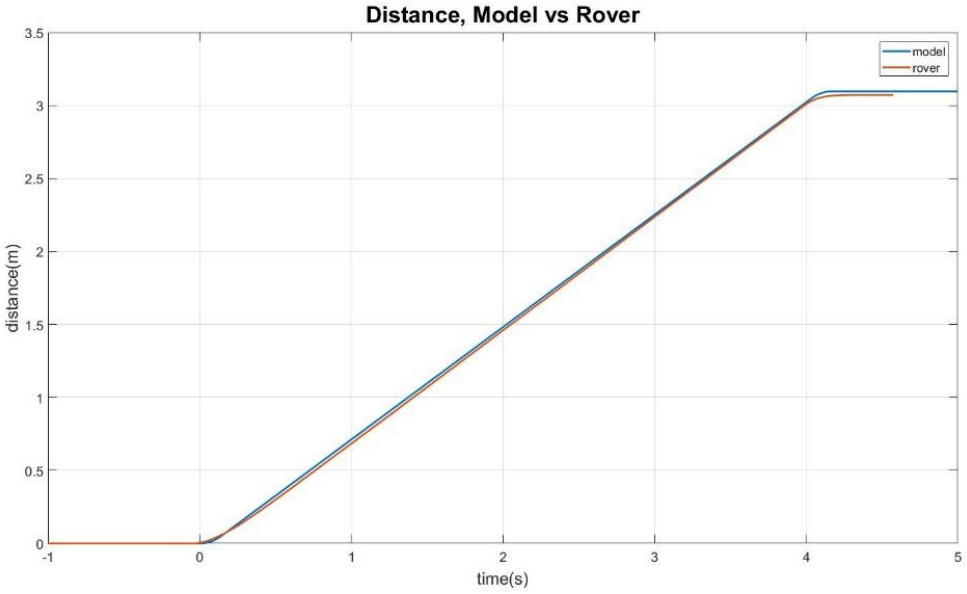


Figure 13. Linear Distance at 0.75m/s

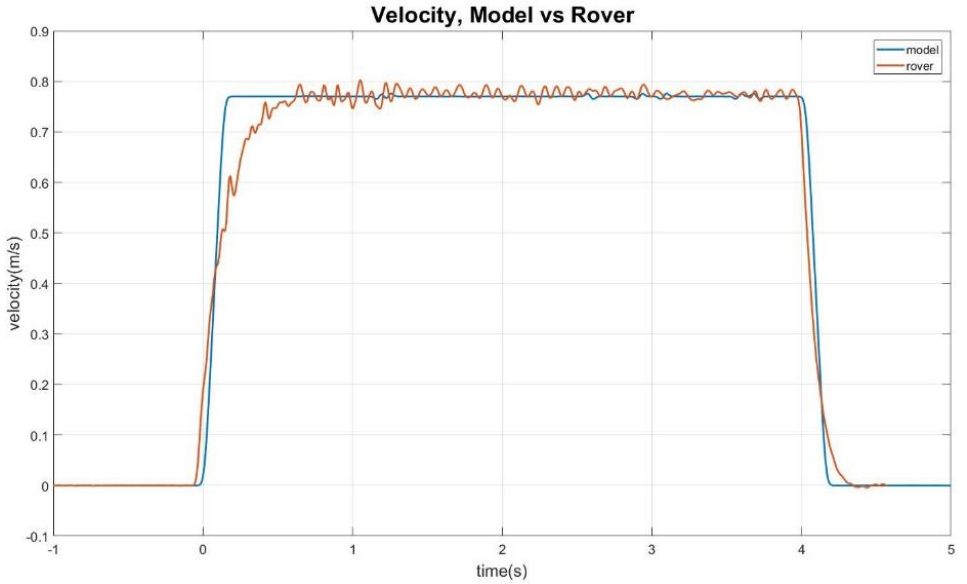


Figure 14. Linear Velocity at 0.75m/s

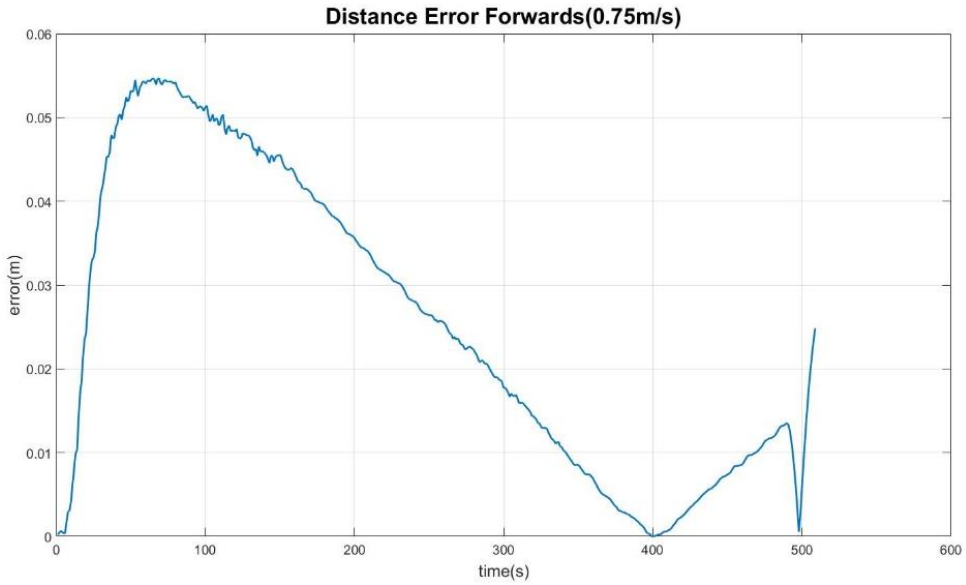


Figure 15. Error in Distance at 0.75m/s

Visually figures 12 and 15 appear to have a significantly different shape, however, their shapes occur for the same reason. The model moves slightly before the rover does (~0.2s). The rover then moves at a slightly higher velocity than the model, meaning that the error (the distance between the model and the rover) decreases. At approximately 4 seconds the rover passes the model and the error again increases. At approximately 5 seconds the rover stops, but the model moves for an additional 0.1 seconds where it overtakes the rover. This causes a rapid decrease in error (the model closes the gap to the rover) followed by a rapid increase in error (the model passes the rover).

Figures 16 to 19 show the results of the third validation experiment from Table 2. In this manoeuvre the robot and the model first move forward in a straight line, to allow the overlaying of both manoeuvres. Secondly, both systems perform a forward right turn. Figure 16 shows the paths taken by both the model and the robot. Figure 17 shows the angular velocity of both the model and the robot over time. Figure 17

shows some noise in the angular velocity of the rover, this is due to the wheels of the rover slipping on the ground surface of the motion capture arena. The slipping was modelled in the simulation environment, but it was unable to replicate the slight sticking of the wheels such as those that occur two and three seconds into the experiment. Figure 18 shows the rotation over time for both the model and the robot. Finally, Figure 19 shows the positional error of the model as compared to that of the robot over time with a mean error of 0.03m. Figure 19 shows two small spikes at approximately 4 seconds, these are due to the rover's wheel slipping slightly.

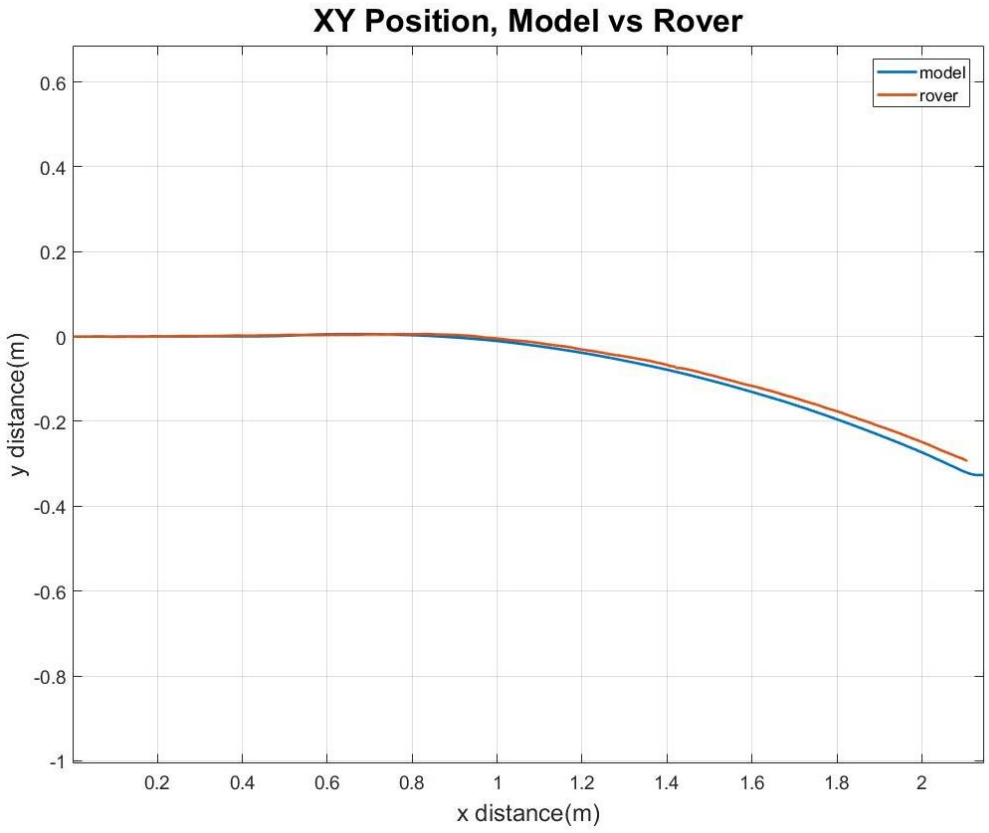


Figure 16. Right Turn Manoeuvre Path

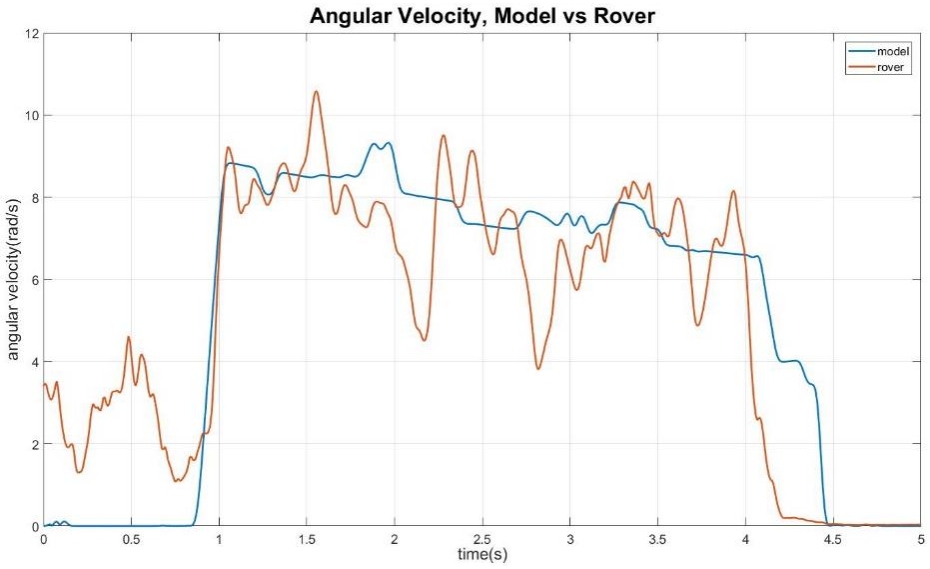


Figure 17. Angular Velocity Right Turn Manoeuvre

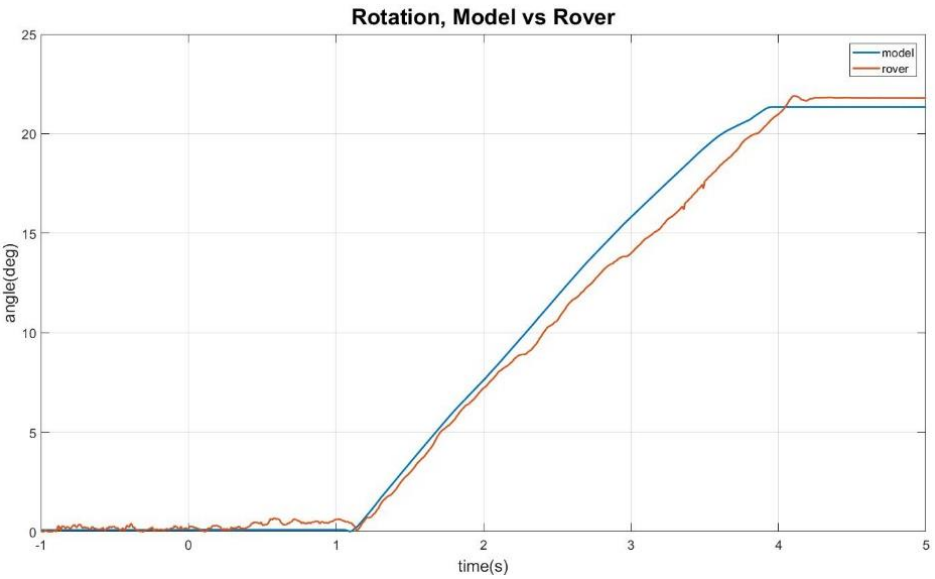


Figure 18. Rover Rotation over Time Right Turn Manoeuvre

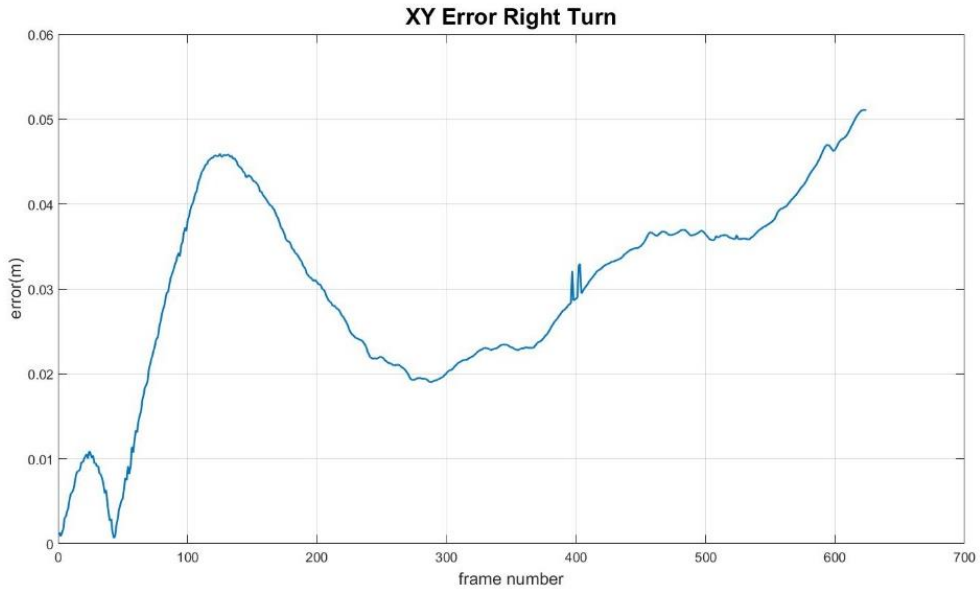


Figure 19. Position Error

Figures 20 to 23 show the results of the fourth validation experiment from Table 2. Again, in this manoeuvre the robot and the model first move forward in a straight line, to allow the overlaying of both manoeuvres. Secondly, both systems perform a forward left turn. Figure 20 shows the paths taken by both the model and the robot. Figure 21 shows the angular velocity of both the model and the robot over time. Again, the rover's plot has several spikes caused by the sticking of the wheels on the rubbered surface of the arena floor. Figure 22 shows the rotation over time for both the model and the robot, this figure shows a more significant rotational error again caused by the inconsistent sticking on the ground surface. Finally, Figure 23 shows the positional error of the model as compared to that of the robot over time with a mean error of 0.0266m.

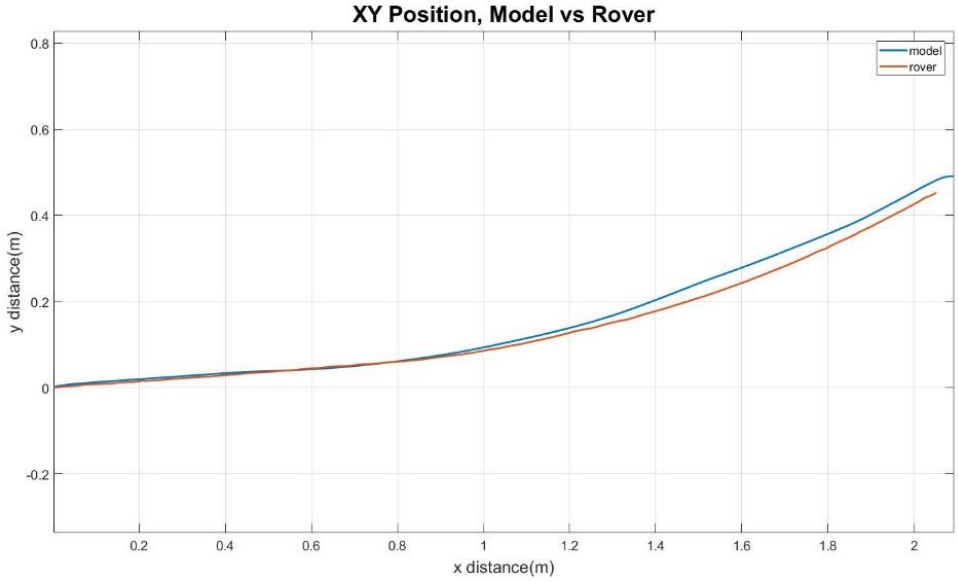


Figure 20. Left Turn Manoeuvre Path

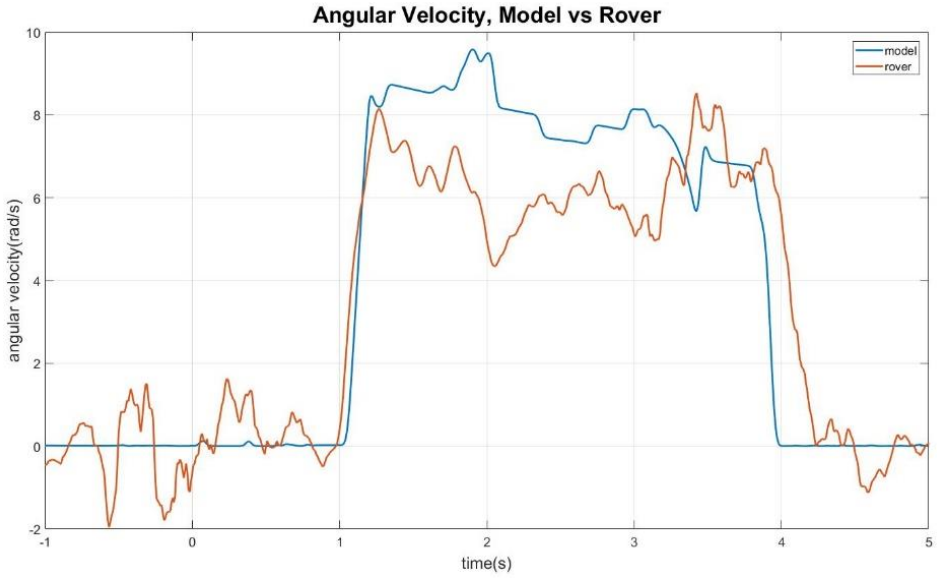


Figure 21. Angular Velocity Left Turn Manoeuvre

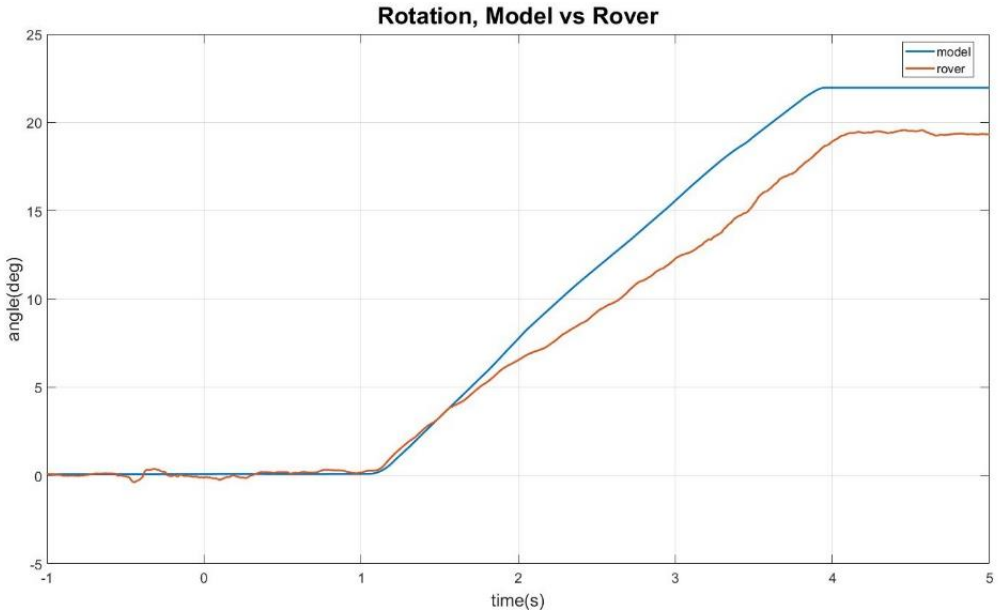


Figure 22. Rotation over Time Left Turn Manoeuvre

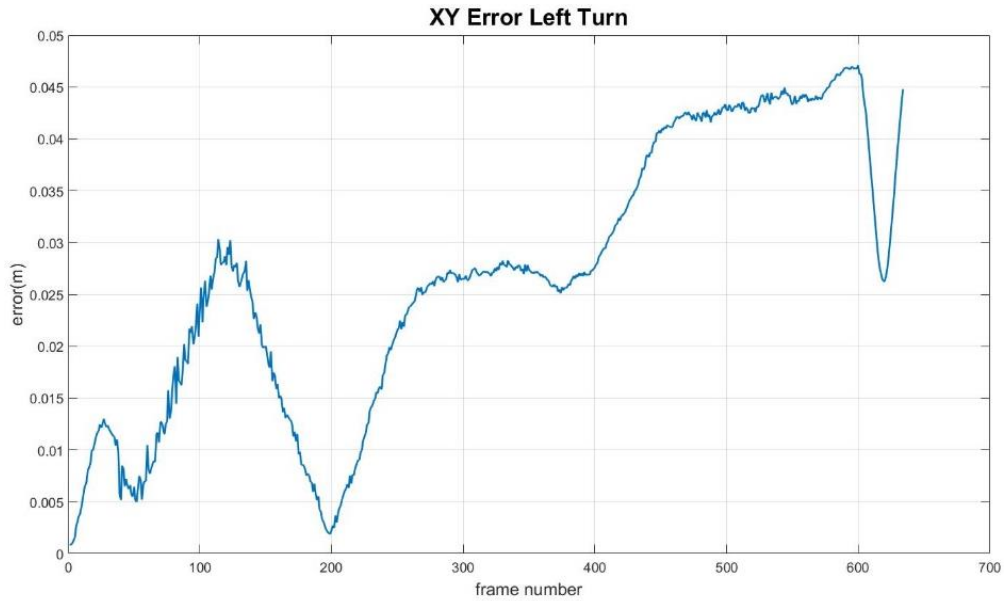


Figure 23. Positional Error

Figures 12 and 15 show a small error between the straight-line distance travelled by the rover and the model. At a velocity of 0.5m/s the mean error was 0.0149m over a distance of approximately 2m (0.745% mean error). At a velocity of 0.75m/s there was a mean error of 0.0253m over approximately 3m (0.843% mean error). This measure of error was calculated as the difference between positions at each time step, this is likely to be the cause of much of the error shown in Figures 12 and 15. The OptiTrack system is accurate to 0.1mm, therefore, the measurements and error estimations are considered accurate enough for the purposes of this research.

Figures 19 and 23 show a small error between the positions of the model and the rover when they are both given the same commands. Around frames 200 to 300 the error increases, this is likely to be caused by the error in rotation over time as seen in Figures 16 and 20. Figures 17 and 21 show that the rotation of the rover is not as smooth as that of the model, this is because at low speeds the rover sometimes sticks in place when operating on the rubber surface of the motion tracking arena. This is also apparent in Figures 17 and 21 where the angular velocity decreases in certain places. These results are not perfect but with the largest error at any given point being 0.045m in a 2.1m manoeuvre (~2% error), the model was deemed suitable for testing purposes. Further attempts to decrease this error were unsuccessful.

3.3 Hardware

The robot used in this research is a Dr Robot Jaguar 4x4 [82]. ROS is used for issuing commands to the robot, as discussed in Section 3.1.3. The controller operates without significant jitter or delay, other than the initial delay in connection as discussed previously. The Simulink based controlled which operates from a desktop computer issues a PWM command via a laptop to the motors of the rover. The controller operates on the error between the desired Tau and the estimated Tau calculated in MATLAB using images from the robot's on-board camera. OptiTrack is a motion

tracking system which has been used during the validation phases of this research, but also for obtaining “truth” data for each of the experiments[42].

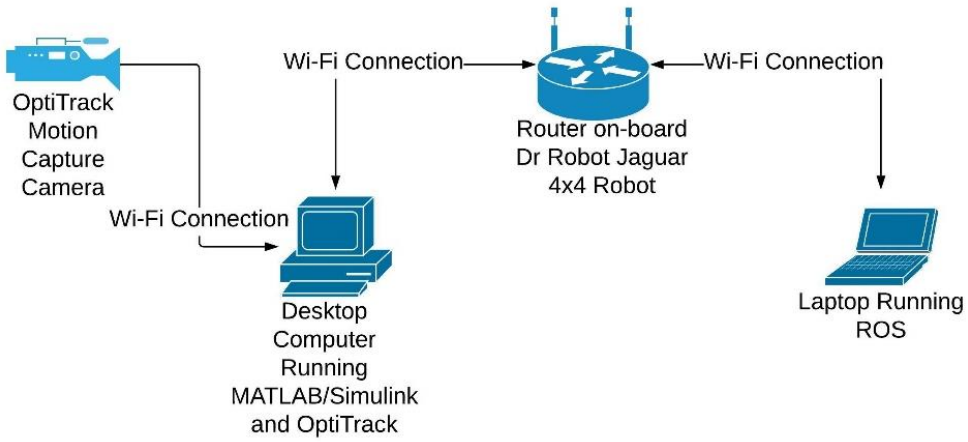


Figure 24. Hardware Implementation

Figure 24 shows how the individual components are connected. The OptiTrack motion capture system is connected to the Windows-based desktop computer via a wi-fi connection to allow for the motion tracking of the rover on the Windows-based computer. The Desktop computer also runs MATLAB/Simulink and is therefore responsible for sending ROS commands to control the rover. The Windows machine sends the ROS commands to the Laptop running ROS via a Wi-Fi connection to the router on-board the rover. This is necessary as the laptop contains the ROS Master which is responsible for tracking Subscribers (receivers of ROS commands) and Publishers (senders of ROS commands) [94], [95]. The laptop then sends the command back to the rover again via the Wi-Fi connection to the router.

3.3.1 Jaguar Rover 4x4

As mentioned previously, the robot used in the hardware experiments is the Dr Robot Jaguar 4x4, as shown in Figure 25. The robot’s software is open source and is compatible with both MATLAB and ROS, which allowed the simple transfer of code

from the simulation environment to the actual robot. The software waits until it receives a ROS message with a PWM command which it then translates to commands to each pair of wheels. The most recent command is executed continuously until it receives the next. There is no on-board processing hardware available on this robot, so both a Linux Ubuntu laptop and a Windows desktop machine were required for the hardware experiments. The laptop running Linux runs the ROS master and a ROS program which publishes topics to provide data from the robot and allowed commands to be published to the robot. The Windows machine ran MATLAB and Simulink in a similar role as it is used in simulation. The windows machine and the Linux laptop are both wirelessly connected to the router which is contained inside the rover, the Windows machine sends ROS commands to the ROS master node on the laptop via the wireless network, which then sends the command to the rover. The Windows machine is responsible for both creating and publishing the ROS commands (it contains the controller) and recording the motion tracking data. One benefit of having two machines is the fact that if the Windows machine is slowed or crashes during the experiment, the laptop can be used to quickly stop the robot.

Table 3 shows the specification for the Dr Robot Jaguar 4x4 [82]. The indoor vision landmark GPS and laser scanner are optional units which were not included in the robot used in this project. These specifications were used to develop the simulation model as discussed in Section 3.1 and therefore the simulation model should closely represent the physical robot.

Category	Feature
Mobility	• Terrain: Sand, rock, concrete, gravel, grass, soil and others wet and dry • Slope: > 45° • Maximum vertical step: 155mm (6”) • Stair climbing: Max stair step height 110mm (4.5”)

	• Traverse: > 200mm (8”) • Speed: 0 – 11Km/hr • Turning radius: 0, min 750mm (29.5”) diameter of turning space • Ground clearance: 88mm (3.5”) • Operator remote control • Autonomous navigation with GPS and 9 DOF IMU (Gyro/Accelerometer/Compass) • Indoor vision landmark GPS (Optional)
Survivability	• Sealed weather resistant enclosure • Temperature: -30° to +40° • Shock resistant chassis • Drop to concrete: Max: 1200mm (4ft) Rated: 900mm (3ft)
Electronics	• Motion and sensing controller (PWM, Position and Speed Control) • 5Hz GPS and 9 DOF IMU (Gyro/Accelerometer/Compass) • Laser scanner (5.6m, 4m or 30m) (Optional) • Temperature sensing & Voltage monitoring • Headlights
Video/Audio	• Colour Camera (640x480, 30fps) with audio
Communication	• WiFi802.11N • Ethernet (Optional)
External Auxiliary Ports	• Ethernet (Optional) • General purpose communication and power port (Optional)

Operator Control Unit	• Gamepad Controller • Head mounted display (Dual 640 x 480), equivalent to 60” display viewed in 2.7m (9 feet) (optional) • Portable computer (Optional) • NVIDIA Shield Gaming Portable Controller(optional)
Power	• Rechargeable battery: LiPo 22.2V 10AH • LiPo battery charger • Nominal operation time: 2 hours (Optional 4 hours)
Motor	• Wheel Motors (24V) : 4 units • Max output (after gear down) (x4): Max 80W, 65Kg.cm/wheel • Rated current: 2.75A, Max current 16A
Dimensions	• Height: 265mm (10.5”) • Width: 573mm (22.5”) • Length: 615mm (24”) • Weight: 20.5Kg (Standard Configuration)
Payload	• Carrying Payload (on flat surface): max 30Kg • (* Upgrade option: max 50Kg, Speed @ 7Km/hr) • Dragging Payload (on flat surface): max 50Kg
Application Development	• Full development kit including SDK, data protocol and sample codes, support Microsoft® Robotics Studio, Microsoft® Visual Studio, ROS, NI LabVIEW®, MATLAB®, Java®

Table 3. Dr Robot Jaguar 4x4 Specification[82]

Figure 25 shows an example of the Dr Robot Jaguar 4x4 robot used in the experimental phase of this project [82]. The two circles on the left are the headlights,

the middle circle is the camera and the component to the right of the camera is the laser scanner. The laser scanner was not installed on the robot used in this project.

Figure 25. Dr Robot Jaguar 4x4

3.3.2 OptiTrack

OptiTrack motion capture equipment [42] was used for two purposes. First, it was used for the validation of the simulation model. Second, it was used for gathering "truth" data from the hardware experiments. Truth data in this sense means the accurate positional data of the rover for the duration of the experiment. For example, a Tau estimation system has been developed as discussed in Section 3.4, the OptiTrack system has been used to record the "True" (actual) Tau value so that the estimation method can be evaluated. The OptiTrack system has a tracking accuracy of 0.1mm and is therefore capable of providing accurate positional data and Tau measurements.

3.3.3 MATLAB

For the hardware experiments, all the scripts were written using MATLAB and therefore MATLAB, running on an external desktop machine, was responsible for

the processing of data and for the issuing of commands to the robot, via ROS. A "master" script was written to ensure that all scripts and other components were synchronized. This included receiving images from the robot, which it processed and estimated TTC as discussed in Section 3.4, below. The control system discussed in 3.1.2 was converted from a Simulink project to a MATLAB script so that they work in the same way.

3.4 Tau Estimation

The method for estimating TTC will be discussed in this section, as well as the implementation and validation of the estimation techniques. OptiTrack was used in the validation of this method, but the TTC value for controlling the model and robot is provided by the TTC estimation method as discussed in Section 3.4.1, they do not use the calculated actual TTC.

3.4.1 Tau Estimation Method

The method of the estimation of Tau used in this research is the expansion method discussed in Section 2.3.1. Equation 11 will be used to estimate Tau, where A is the area of the blob detected in images from the camera or simulated images and $\dot{A}$ is the rate of change of A with time. At each iteration of the estimation, the A is measured and the $\dot{A}$ calculated using the time-step. A moving average is applied to $\dot{A}$ to remove spurious estimations which can have a negative impact on the Tau estimation. Tau is then estimated using Equation 12.

In the hardware implementation of the Tau estimation method, an extra step is required. As the real world is much more noisy than a simulation environment it is necessary to take extra steps to identify the obstacle. The obstacle was covered with red paper and other red items in the background were removed. A filtered image is created where the red remains as it is and everything else is covered with black, as shown in Figure 26. This is achieved by setting acceptable values for Red Green and Blue (RGB) in the image, pixels that are within the acceptable levels remain as they

are, everything else is changed to black. This makes it much easier to detect the obstacle as it is the only blob in the image, however, outside of the lab setting this would be a problem as there is likely to be more than one object in an image and there are a variety of different coloured object. With this previous point considered, this solution is deemed suitable for lab testing. Figure 26 shows an obstacle that has been extracted from the camera images using a red colour extraction which is then converted into a binary image, as shown in Figure 27. In simulation, using the known height and width of the obstacle and the known focal length of the camera, an equivalent binary image is generated in Simulink, as shown in Figure 28.

Matlab and Simulink were chosen to implement this project due to their ease of development. Both pieces of software provide a number of tools such as blob detection and filtering allowing for a more rapid prototyping. It could be argued that this software is not as fast as languages such as C or C++, however, this would have slowed the development of the project. It is recommended that future researchers and developers should attempt to implement this project in a faster programming language. Using a faster programming language would allow for more frames to be recorded per second which may provide a more accurate estimate of Tau, as discussed in section 3.4.2.

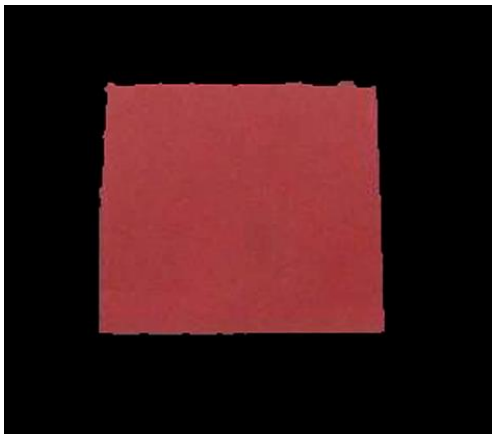


Figure 26. Red Obstacle Extracted

Figure 27. Image converted to Binary

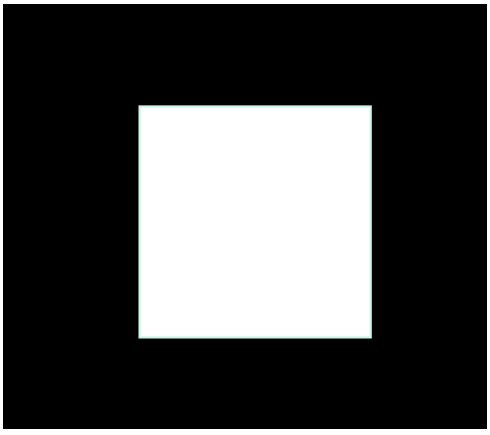


Figure 28. Simulated Image

3.4.2 Tau Estimation Validation

This subsection will discuss the validation results of the Tau estimation system that was implemented as well as some of the problems that were encountered on the way. Firstly, it was discovered that this method of estimating Tau is dependent on a high frame rate. The higher the frame rate the more accurate the estimation as can be seen in Figure 29. A frame rate of at least 3 frames per second (FPS) is required to achieve a Tau estimation accurate enough to be used to control a robot. An error of approximately 1 second can be tolerated, though it is more important that the estimation is smooth to allow smooth control. To quantify the effect of changes in frame rate, a simulated experiment with calculated image size over time at different frame rates was performed. Figures 29 to 32 show that with the higher frame rate there is less error between the estimated and the actual Tau. Figures 29 to 32 also shows what happens when the object fills the entire image, at approximately 9 seconds the object fills the screen and the system is unable to estimate Tau from this point on as $\dot{A}$ becomes zero.

Figure 29 shows the results with 3FPS and Figure 31 shows the same experiment allowing 10 FPS. Figure 30 shows the error for the 3FPS experiment which is greater than that of the 10 FPS experiment shown in Figure 32. The "actual Tau" is the truth data that is calculated using the known position and known closing velocity. To reduce the effect of this problem equation 18 was derived to estimate the error induced by the frame rate, which was then subtracted from the estimation. The proof for equation 18 can be found in the Appendix A of this thesis.

$$E = \Delta t - \frac{2\tau\dot{x}\Delta t}{2\dot{x}\Delta t + 8\tau\dot{x}} \tag{18}$$

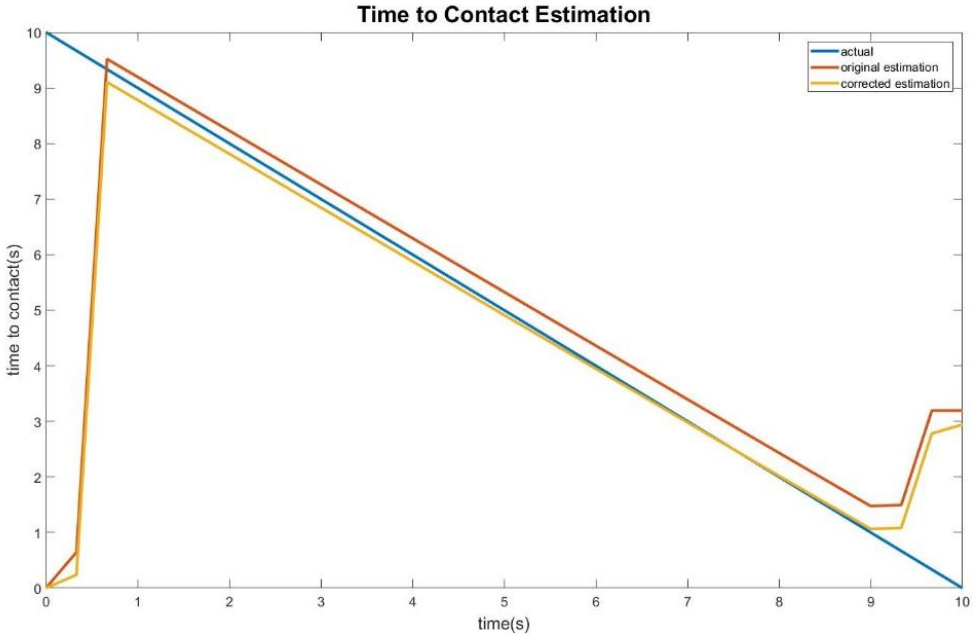


Figure 29. Tau Estimation (3FPS)

Figure 29 shows the original estimation in red, there is a small but obvious offset in the estimation which will always be present when the time-step is greater than 0. When estimating the error using equation 18 and subtracting this from the original estimation the error is significantly reduced.

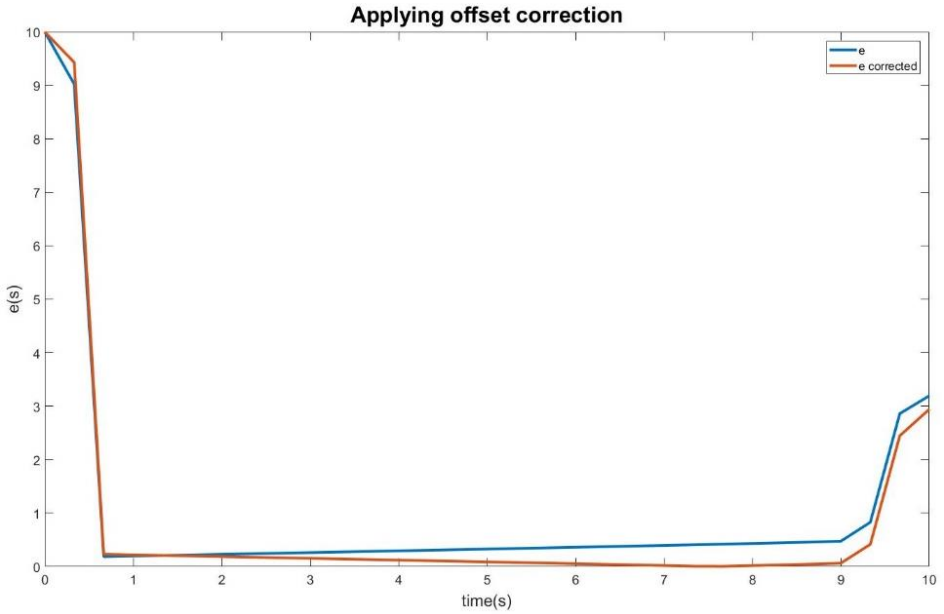


Figure 30. Tau Error (3FPS)

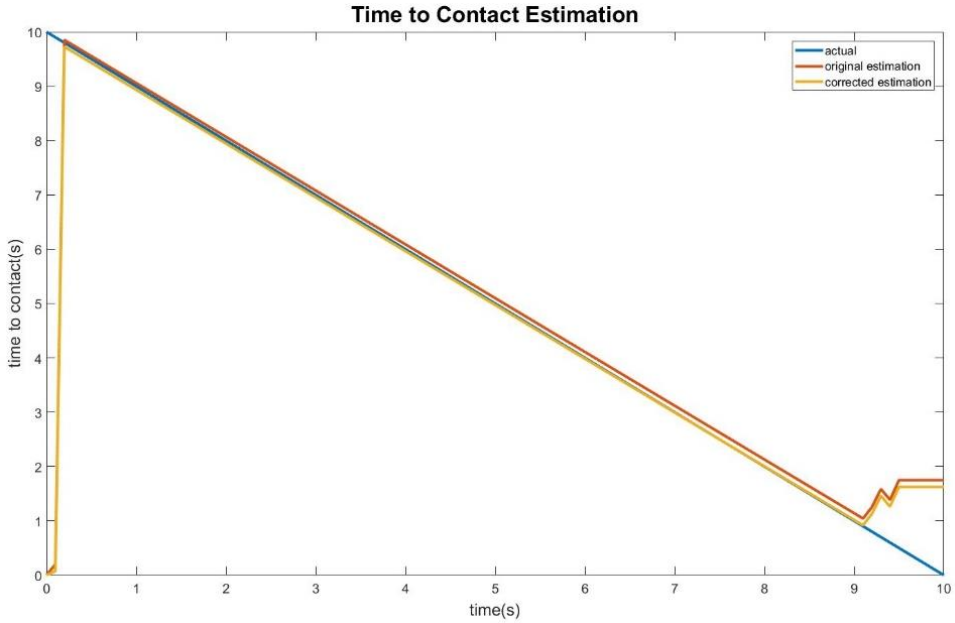


Figure 31. Tau Estimation (10FPS)

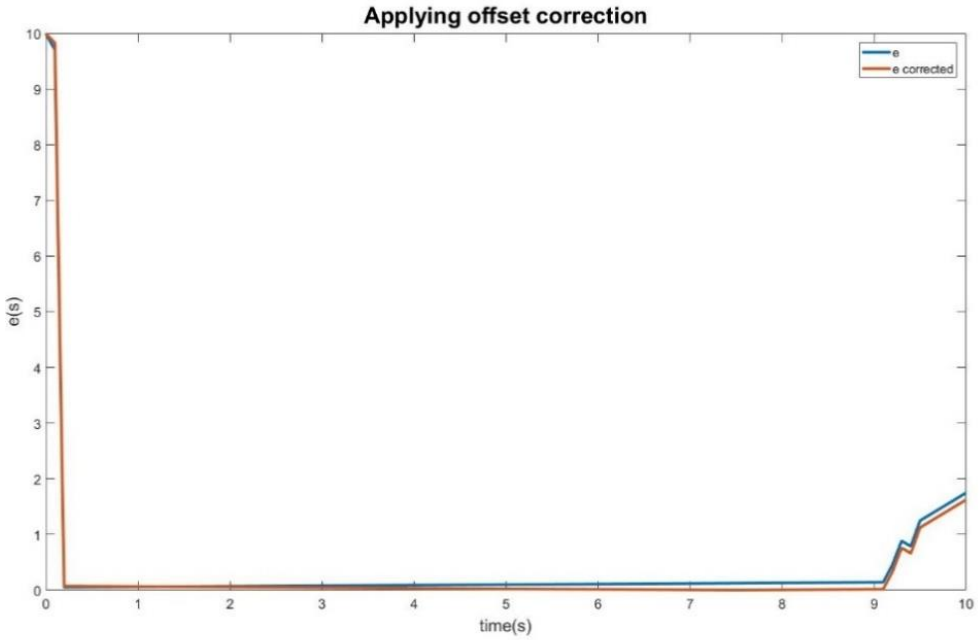


Figure 32. Tau Error (10FPS)

When performing real experiments, even with the simplified backgrounds and the red object light conditions can still affect the obstacle detection especially around the edge of the image. This prevents the identification of a perfectly rectangular object with perfectly straight edges. This affects the area measurement of the object in the image and therefore the measurement of the rate of change of the area, thus ultimately inducig an error into the Tau estimation.

Figure 33 shows an example of an unfiltered measurement of the area of the object, compared to the mathematically calculated area. The curve is not perfectly smooth. This results in a noisy calculation of the rate of change of area as shown in Figure 34. $\dot{A}$ is an integral part of the calculation to estimate TTC and with a measurement as noisy as that of Figure 34, it is impossible to obtain a usable TTC estimation.

Therefore, it is necessary to implement some noise filtering for the area measurement.

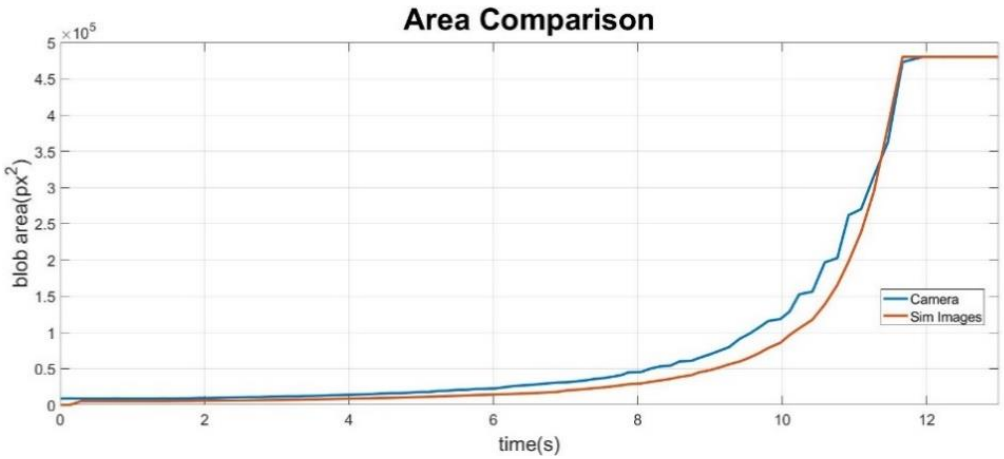


Figure 33. Area Measurement

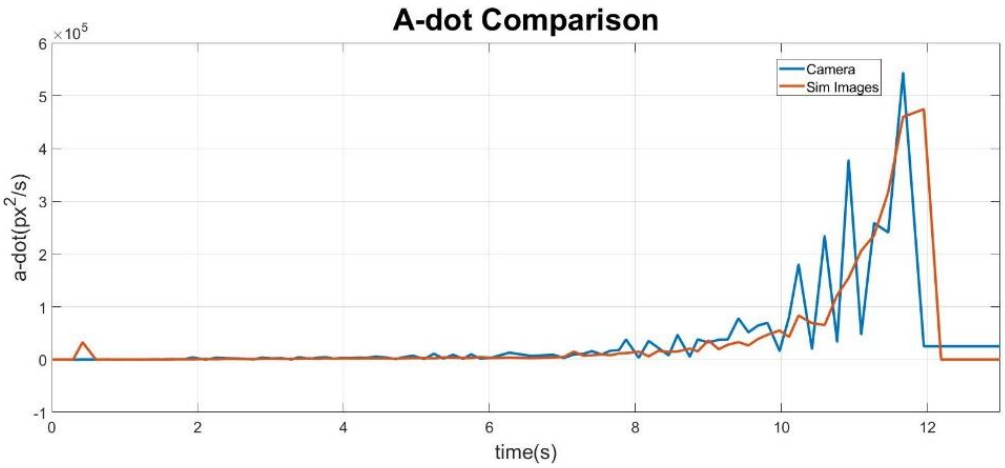


Figure 34. Rate of Change of Area

Filtering by implementing moving averages can increase the offset if too much is applied as shown in Figures 35 and 35. Figure 35 shows the area measured without

filtering, it is clear that it is not smooth but there is little error in the measurement. Figure 36 shows a smoother area measurement, but the error of the measurement is increased. Although the area measurement in Figure 36 would provide a smoother TTC estimation, it would introduce a large time delay. It is therefore important that minimal filtering is applied as too much can decrease the area measurement and introduce a time delay. Although it can be tempting to also filter $\dot{A}$ this can introduce even more delay and should therefore be avoided.

The system was able to operate at approximately 10FPS, this is significantly above the 3FPS required to provide a reliable estimation of Tau. It therefore appears that the time taken to process the data does not significantly affect the estimation. No obvious effects of operating system jitter were experienced.

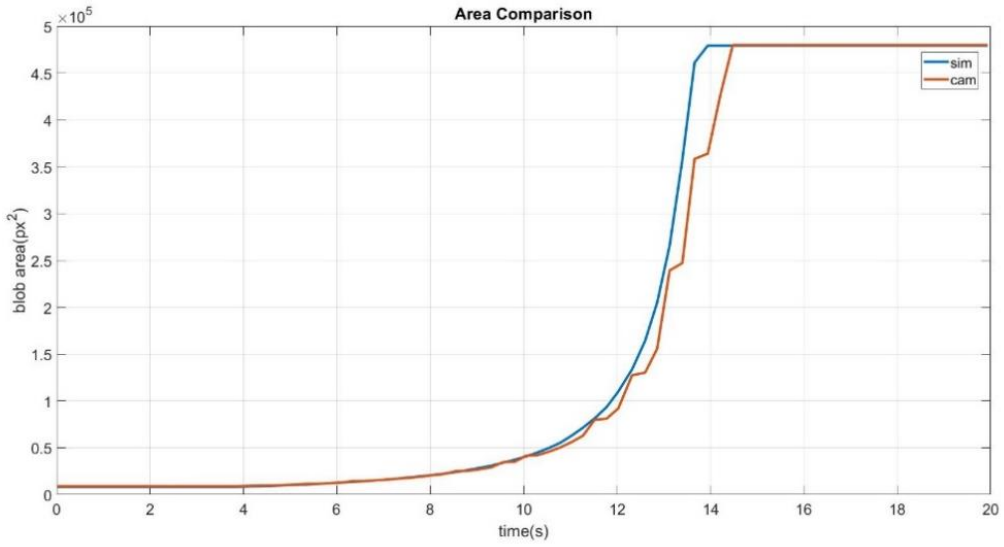


Figure 35. Unfiltered Area Blob Measurement Comparison

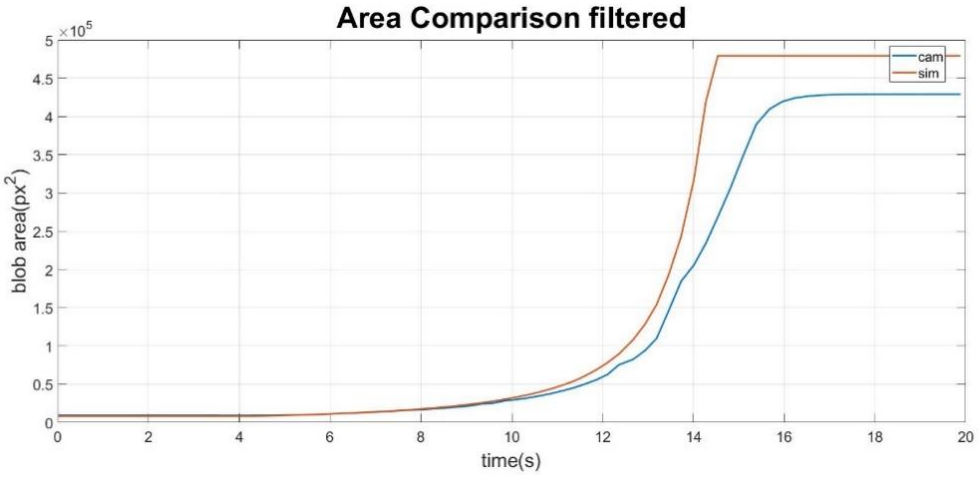


Figure 36. Filtered Area Blob Measurement Comparison

With a combination of minimal filtering and the tuning of the colour extraction method, a much smoother area measurement was achieved as shown in figure 37. To find the minimal filtering levels the TTC estimation experiment was repeated starting from zero filtering and increased until a value was found that produced a smooth TTC estimation. Figures 38 and 39 show minimal error between the area that was measured and the mathematical calculation of what the area should be based on geometry. This measurement allows for a usable estimation of TTC to be generated.

Figure 37 shows a slight error between the camera and simulation at a gap of 1.4m. This is due to the blob detection method; it has not been able to refresh and measure the expansion of the blob and has therefore returned the previous measurement. This is evident in the errors shown in figures 38 and 39. This occasionally occurs and can affect the simulation and hardware experiments, but the level of error is tolerable for the requirements of the experiments performed in this thesis.

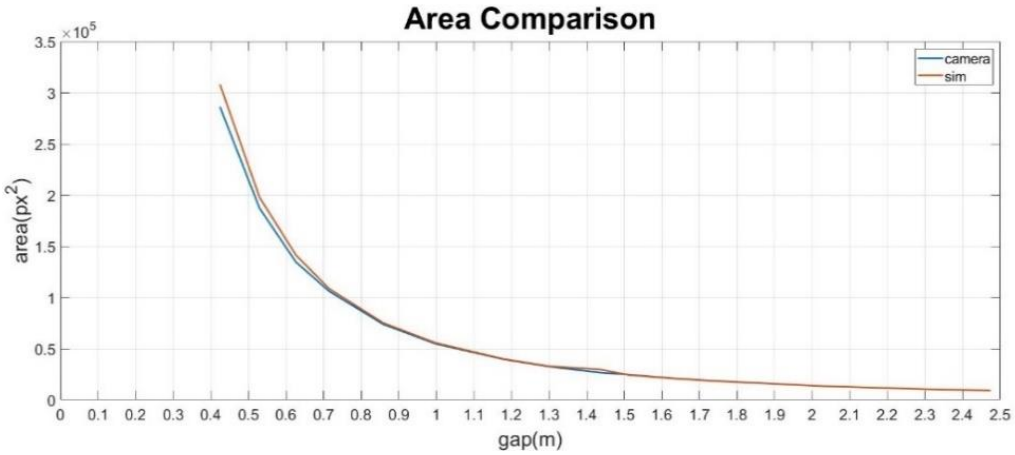


Figure 37. Blob Area Measurement Comparison

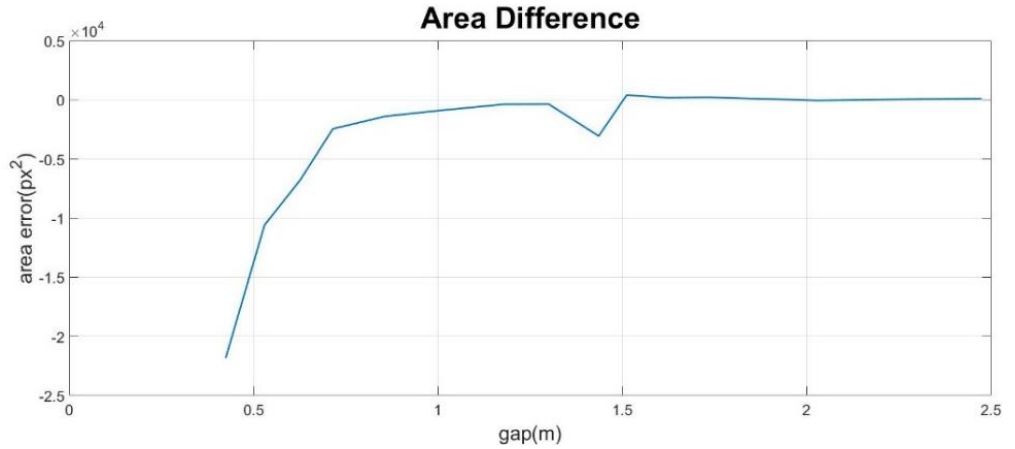


Figure 38. Area Error (px^2)

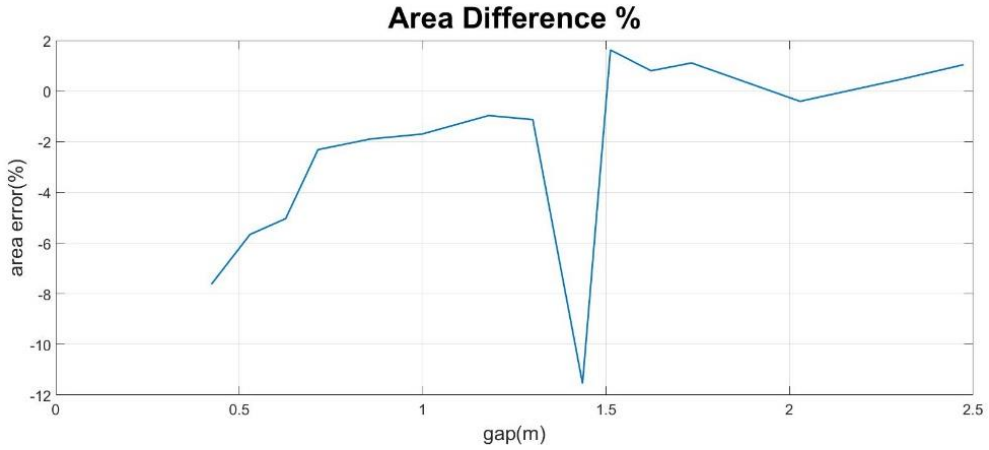


Figure 39. Area Error (%)

Figure 40 shows the TTC estimation that was achieved using the same area measurement method used to obtain Figure 37. The actual measurement is obtained using the OptiTrack system to measure the gap between the robot and the obstacle over time. The offline estimation was produced using mathematical measurement based on geometry. The real time estimation is the estimation based on the camera-based estimation discussed in this chapter, this is the measurement that is to be used in the experiments. It is clear from Figure 40 that even after filtering the area, TTC is still not smoothed enough to be used to control the robot. This effect is because blob detection is often not perfect especially in daylight where the brightness of the image may change during an experiment. It is therefore necessary to further filter the final TTC estimation.

Figure 40 shows that for the first 4 seconds the estimation is very noisy, this is due to the rover starting from a standstill, while the rover is stationary Tau is infinite. At approximately 4 seconds an estimation of Tau is achieved but it is not until approximately 9 seconds that an accurate estimation is achieved, this is again due to the blob detection method and a noisy area measurement.

At approximately 1.4m there is a significant drop in the area difference and increase in the error. This is due to filtering technique applied, occasionally the rover may slip or stick, and this may be seen as a spurious data point in terms of filtering, hence the increased error for this point only.

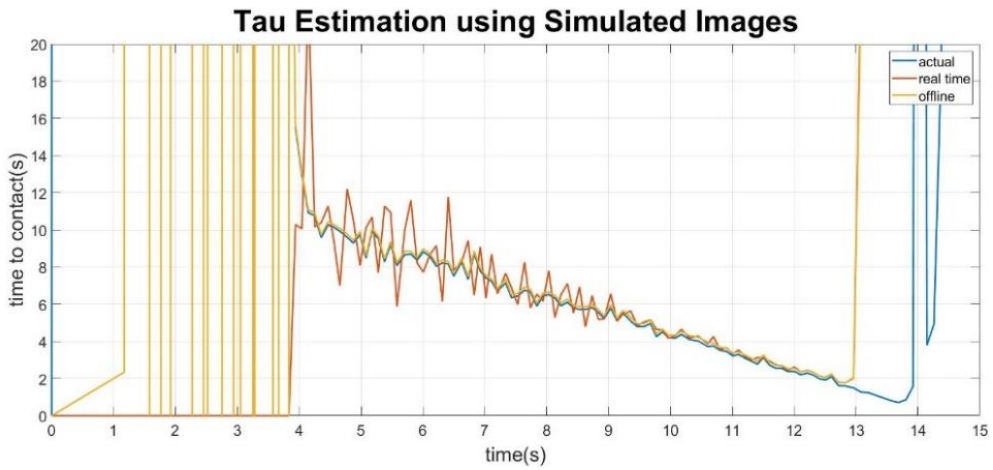


Figure 40. TTC Estimation Unfiltered

Figure 41 shows the result of the TTC estimation after moving average filtering. The real time estimation is now much smoother but an offset of up to one second has now been introduced. The benefit of the smooth estimation outweighs the disadvantage of the offset caused by the filtering. To compensate for the offset of the estimation, all manoeuvres will be programmatically stopped when the object fills the image as this is the point where a Tau estimation is no longer achievable. In Chapter 4 it will be shown that when using this expansion method of estimating TTC it is not possible to estimate Tau once the object completely fills the image. It is for this reason that all manoeuvres will be stopped when the image is saturated, i.e. the rover will apply full brakes.

As shown in Figure 41, for the first 4 seconds there is no estimate or actual Tau, this is because the rover started from a standstill. When the rover is not moving Tau is infinite. At approximately 4 seconds a Tau estimation is achieved this becomes more accurate at approximately 7 seconds as the rover gets closer to the obstacle and can measure its area more accurately. An accurate estimation is maintained until at 14 seconds it is lost as the obstacle completely fills the image and the rover therefore brakes to avoid collision. The braking at this point causes the actual and estimate Tau recordings to increase to infinity. The estimation has an almost constant offset (~1s) due to the filtering that is applied to the area measurement, this error can be tolerated for the purpose the manoeuvres presented in this thesis. An over estimation is considered dangerous as the rover is closer to the obstacle than the estimation suggests, however, the automatic braking system halts the rover as soon as the image becomes filled by the obstacle meaning that the rover stops before it hits the obstacle, regardless of the estimation.

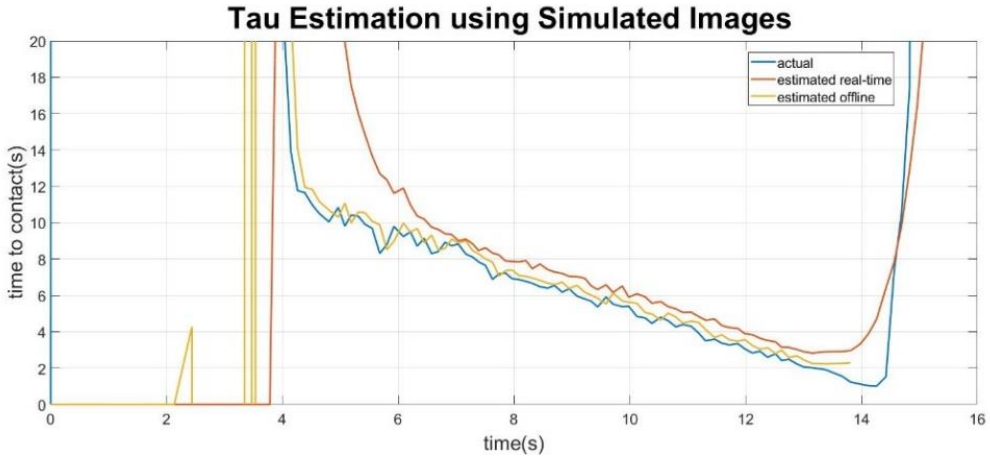


Figure 41. Filtered TTC Estimation

3.5 Conclusion

This chapter has discussed the methodology used in research phase of this project. The Gazebo based simulation environment has been developed to model the hardware implementation of this project. A Dr Robot 4x4 has been used for the hardware experiments, this is controlled via a computer running MATLAB/Simulink and a laptop running ROS. The OptiTrack system has been used for motion tracking to enable validation of the simulation environment and for the recording of positional data for hardware experiments. The results shown in this chapter show that the simulation environment closely matches that of the hardware implementation. It also shows that the optical expansion method of estimating Tau produces a smooth and accurate estimation with an acceptable offset caused by filtering. Finally, this chapter introduced a method of removing the offset which occurs due to the timestep to enable a more accurate estimation which is less affected my larger timesteps.

Chapter 2 discussed the cutting-edge CSAIL system, one of the limitations of this system was that it did not sense objects in its path for the first 10 metres of its flight or for the 10 metres following a sharp turn. The Tau-based perception method discussed in this chapter improves on this limitation, Tau is constantly sensed and therefore obstacles are sensed regardless of the phase of the manoeuvre.

CHAPTER 4

LINEAR MANOEUVRE EXPERIMENT

In this chapter the first manoeuvre will be discussed. This experiment simulates a driver stopping before an obstacle or a stop sign. This chapter will describe results for both a simulation and a hardware manoeuvre. In the development of this manoeuvre several different methodologies were used to build up to the final experiment. Firstly, a simulation experiment was performed using only ground truth TTC data, secondly the ground truth TTC data was replaced with the TTC estimation of Section 3.4.2. Finally, the experiment was performed in hardware. This multi-step approach allowed for effective debugging of the system.

Chapter 3 discussed the simulation and hardware used for this experiment. The hardware consisted of a Windows-based computer for OptiTrack and MATLAB Simulink, a Linux Laptop for publishing ROS commands to the rover and a router on-board the Dr Robot Jaguar 4x4 robot, all of which were connected via a Wi-Fi network. The robot is controlled by MATLAB which sends PWM commands using ROS, in the hardware experiment these commands are sent over a Wi-Fi connection. Figure 24 shows how all these hardware components are joined.

In this experiment, there is an obstacle in the path of the rover, the robot must stop before it impacts the obstacle. Figure 42 shows an example of an experiment performed, there is an obstacle in front of the rover and the aim is to close the gap in x while also preventing a collision. Note that the checked image attached to the top of the box is not used in any experiments related to this project. The constant $\dot{\tau}$ approach is used to decelerate and stop the rover as discussed in section 2.2.2. This

approach is based on research into how human drivers decelerate and stop their vehicles [31], [34], [35]. As discussed in Chapter 2 Tau guidance involves keeping one parameter in proportion with the Tau of another, using a coupling term (K) [98]. If the coupling constant K is maintained at 0.5, the rover will stop exactly at the obstacle, if K is larger there will be contact with the obstacle, and if K is smaller and the rover will stop short. For this experiment a K value of 0.5 is the target, which is the same as what human drivers are found to also maintain [31].

As the Tau estimation method discussed in chapter 3 relies upon the measurement of the rate of change of the area of the obstacle, a reliable estimation of Tau will not be possible when the obstacle fills the entire image taken by the rover's camera. When the obstacle completely fills the image, it triggers a function which stops the manoeuvre and causes the rover to brake. For the obstacle used in this experiment, this has led to the robot stopping approximately 0.3m before the obstacle, the point at which the robot's camera image is saturated by the obstacle. This trigger ensures that the robot continues the desired manoeuvre for as long as possible but also stops safely when it is unable to estimate Tau as shown in figures 45 to 49.

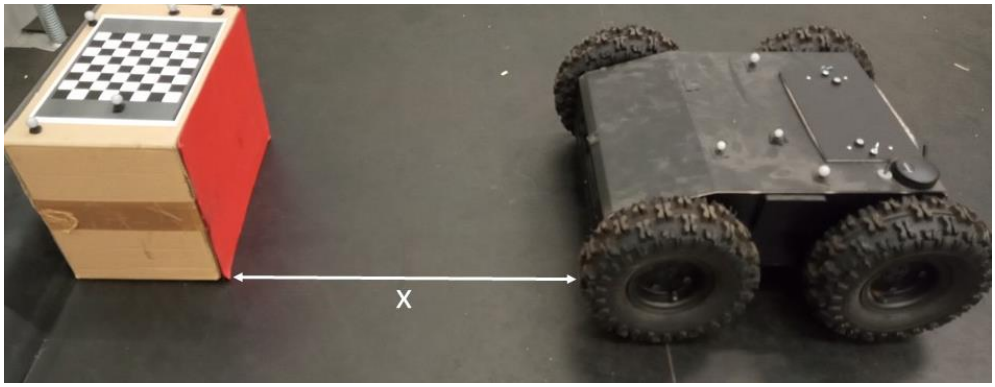


Figure 42. X Gap Hardware

4.1 Constant Tau Dot

In chapter 2 Constant Tau Dot guidance was discussed, Figures 5 and 6 show the form that they take. In this experiment TTC is constantly estimated and once its estimated value becomes lower than a user-defined safe threshold a deceleration manoeuvre is triggered. Once triggered, the manoeuvre time T is set at the current TTC allowing for a safe and smooth manoeuvre, this prevents large initial decelerations. K is set at 0.5 as this is the value shown observed in human braking experiments, it is also the value which ensures that the robot will stop before colliding with the obstacle [34]. Using these values of K and T the TTC profile was generated, this was then regarded as the "desired TTC" and a Proportional controller was then implemented with the aim of ensuring that the robot's actual TTC matches this desired TTC. The aim of the controller is to minimise the error between the desired Tau and the actual or estimated Tau. The controller is shown in equation 19 and figure 43 for simulation and figure 44 for hardware.

Figure 43 shows the controller for the simulation experiment and Figure 44 shows the controller for the hardware experiment. The main difference between these diagrams is the fact that the simulation controller uses simulated images to estimate TTC and the hardware experiment uses real-time images from a hardware camera to estimate TTC. Simulated images were used to simplify the simulation environment and to avoid known and at the time unsolved issues with Gazebos camera system. In Figures 43 and 44 τ_d is the desired TTC and τ_e is the estimated TTC.

Equation 19 shows the equivalent of Figures 43 and 44. This equation shows that a proportion of the error is applied to the current velocity command μ_n. If $\tau_d - \tau_e = 0$ then $K_p + 1$ allows for the continuation at the previous velocity command μ_{n-1}.

$$\mu_n = \left(\left(1 - \frac{\tau_d}{\tau_e}\right)K_p + 1\right)\mu_{n-1} \tag{19}$$

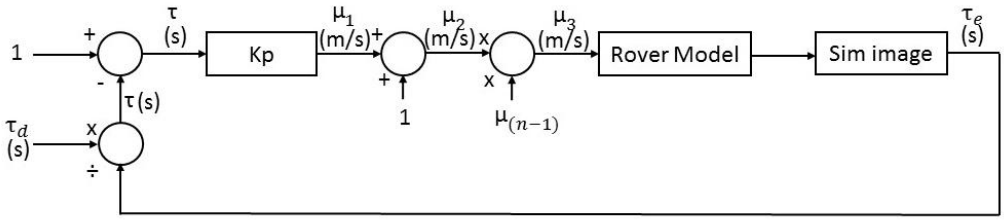


Figure 43. Control Scheme used for Simulation Experiment

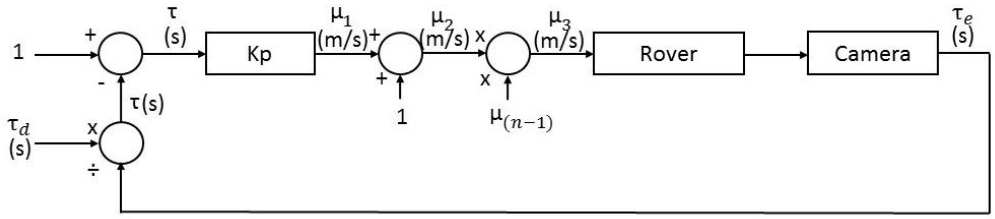


Figure 44. Control Scheme used for Hardware Experiment

4.2 Simulation Experiment

In this experiment an obstacle is placed 20 metres in front of the model rover. A P controller produces the linear velocity based on the error between the desired time to contact and the actual time to contact which is generated using constant $\dot{\tau}$. The desired time to contact is produced using constant $\dot{\tau}$ and the actual time to contact is calculated in the Gazebo simulator using the actual distance between the model and the obstacle over the model's velocity.

4.2.1 Truth Data

In this initial experiment there is no estimation of TTC, instead of the estimated TTC being used in the controller shown in Figure 43, the actual TTC, measured using OptiTrack, is used as the input. Subsection 4.2.2 will discuss the extension of this experiment to include the use of estimated TTC.

Figures 45 to 47 show that the distance travelled by the simulated rover starts at a non-zero value, this is because the constant $\dot{\tau}$ manoeuvre requires a non-zero velocity to work and so the robot must already begin closing the gap before the manoeuvre is triggered. Figures 45 to 47 show that the model had an initial TTC much higher than desired therefore initially the rover must accelerate to reach the desired TTC. Figures 45 to 47 shows that once this point has been reached the robot decelerates at a constant rate. Figure 45 to 47 shows that there is initially a large error which takes less than one second to reduce, this is the time that the controller takes to get the robot back to the desired value of $\dot{\tau}$.

It is clear from the Figures 45 that there is a constant deceleration of the model as would be expected from this Tau guide when K = 0.5. Figures 45 to 47 show that τ reaches zero when time reaches zero, as expected. Figures 45 to 47 shows that in this experiment there is a τ error whilst the model accelerates to reach the target τ, then there is zero error. In the experiment there must always be an initial velocity which may not always be the desired initial velocity. This means that it is unlikely that the desired and actual velocity will match and that there is usually an initial acceleration or deceleration. This phase appears before the expected constant deceleration as can be seen in Figure 45 when K = 0.5. Figures 45 to 46 shows that the model reaches the obstacle (20m away) when the time reaches zero.

This experiment was repeated with K = 0.3 as shown in Figure 45, K = 0.5 as shown in Figure 46 and K = 1.0 as shown in Figure 47. As discussed previously a K = 0.5 is most commonly observed for human drivers, this provides a smooth constant level of deceleration where as a K of 0.3 provides a much harsher deceleration as seen in Figure 45, this manoeuvre still manages to stop before colliding with the obstacle, it stops slightly earlier than when using K = 0.5. In the last 8 seconds of the manoeuvre the robot very slowly moves the remaining few centimetres as shown in Figure 45. This is a less natural appearing manoeuvre than K = 0.5 which closes a similar distance in approximately 4 seconds as shown in Figure 46.

Figure 47 shows that with K = 1.0 there is no deceleration, the robot continued at a constant speed of approximately 0.9m/s and collided with the obstacle. At the very end of the manoeuvre at time 0, the PWM commands begins to increase as shown in figure 47. This is because the gap has closed to zero as the robot collides with the obstacle, but then the gap increases and so the controller attempts to close this gap again.

Figures 45 to 47 also show the effect of different values of K on the desired Tau profiles. Each desired profile will provide a constant rate of change of Tau until Tau is closed to zero at time zero. Each of the experiments shown in figures 45 to 47 where given a T of 20, which means that the manoeuvre will last for 20 seconds. The desired Tau at the beginning of the manoeuvre starts from T * K which means that the desired Tau for K = 1 starts from 20 seconds as shown in Figure 47, 10 seconds for K = 0.5 as shown in Figure 46 and 6 seconds for K = 0.3 as shown in Figure 45. This can result is an initial acceleration or deceleration phase which is shown in the first 1 to 2 seconds of the manoeuvres as shown in Figures 45 to 47.

For the remaining experiments, a K of 0.5 was chosen to ensure safe hardware experiments and the most realistic human-like braking strategies [31]. As mentioned previously an additional braking strategy will also be used which will ensure that the robot will apply full brakes to stop as soon as the image from the robot's camera is completely filled with the obstacle.

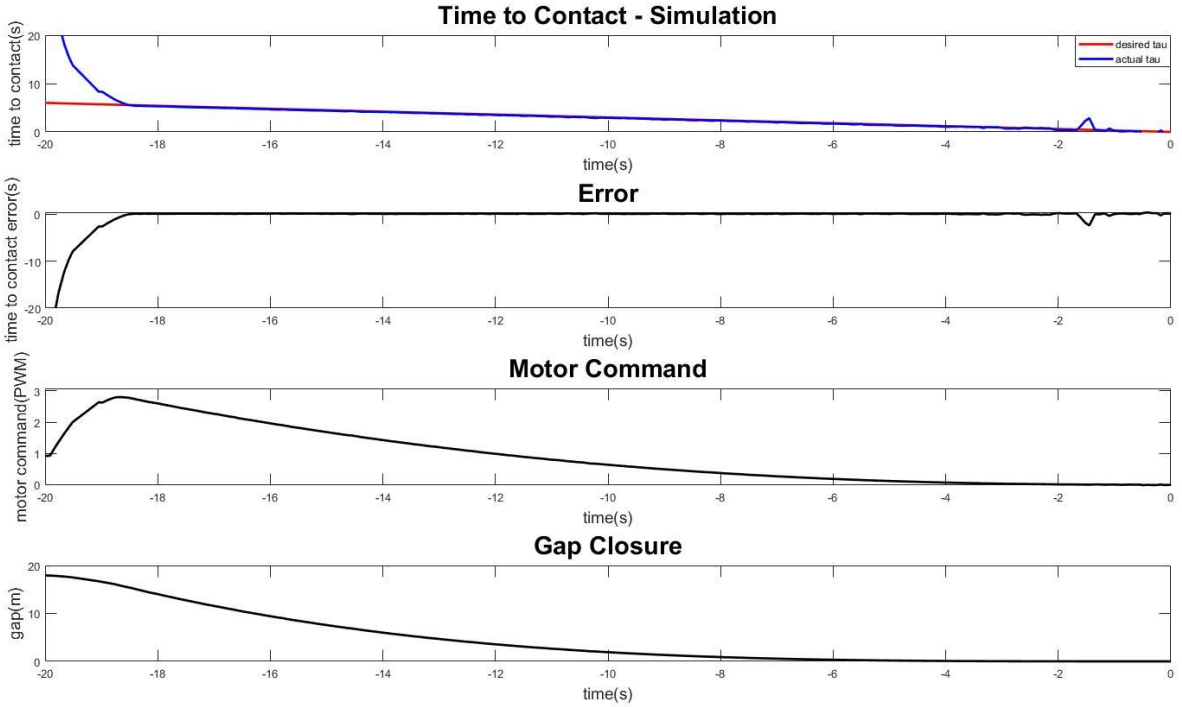


Figure 45. Results obtained in the Simulation Experiment for K = 0.3

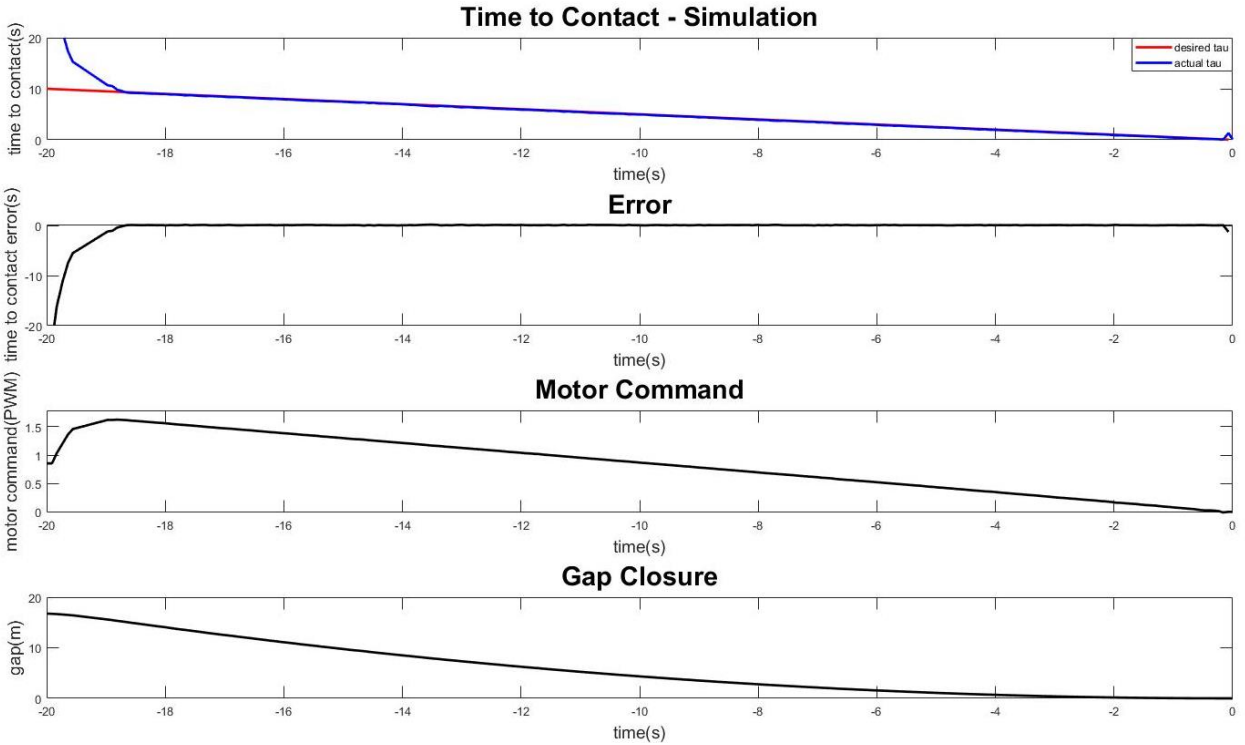


Figure 46. Results obtained in the Simulation Experiment for K = 0.5

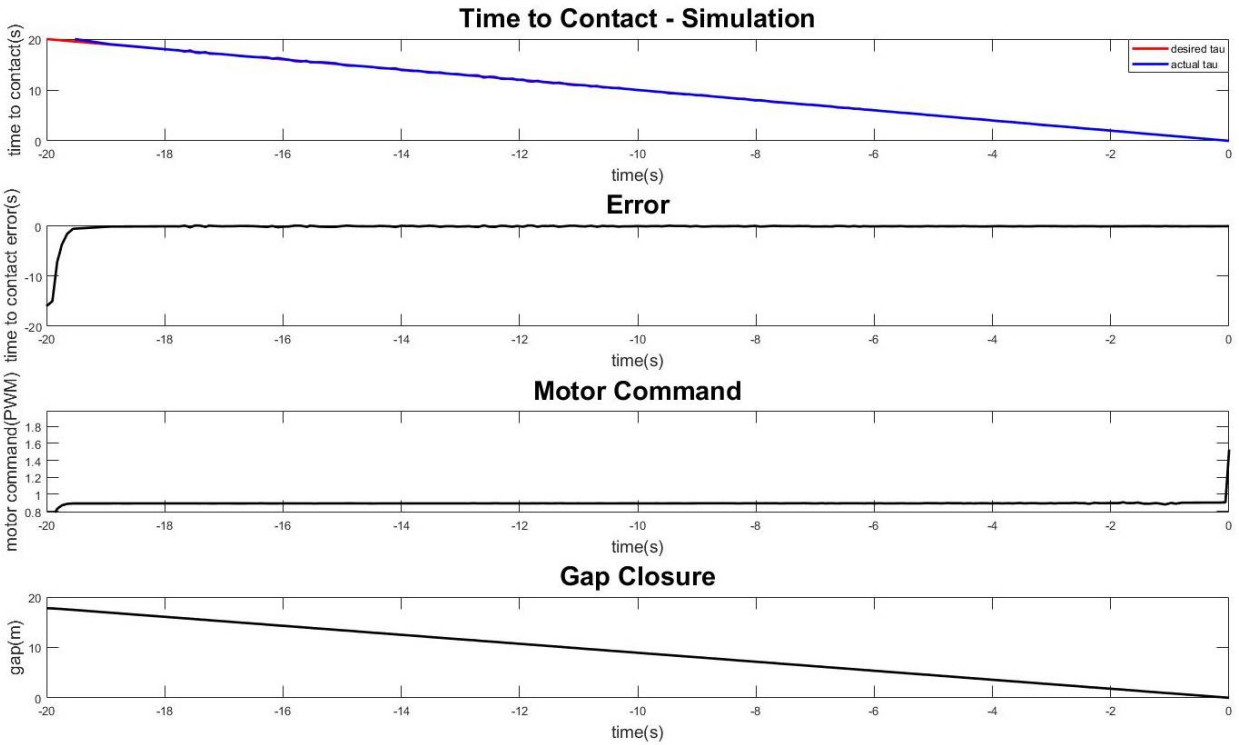


Figure 47. Results obtained in the Simulation Experiment for K = 1.0

4.2.2 Estimated TTC

Figures 45 to 47 show the results of the linear experiment manoeuvre using truth data from the simulation environment. This useful for testing the proposed Tau-based guidance and for tuning the controller. However, this is not a realistic experiment, in the real-world autonomous systems are unlikely to have access to external positional data, therefore, it is necessary to estimate TTC on-board.

Figure 48 shows the results of the experiment using estimated TTC instead of the truth data. The error between the desired and actual TTC is significantly larger. This is to be expected when using an estimation as opposed to truth data as shown in Chapter 3. Regardless, the experiment was a success and the gap were successfully closed at a constant $\dot{\tau}$. The experiment was considered a success as both the velocity and linear gaps were closed whilst maintaining a constant $\dot{\tau}$ with minimal error between the desired and estimated $\dot{\tau}$. The results shown in Figure 48 can therefore be considered as displaying human-like braking [34], [35] which as discussed in Chapter 1 may aid the acceptance by other road users [23], [24].

Using the expansion method of estimating TTC, it is necessary to stop the manoeuvre early. This is because when using this method, it is impossible to estimate TTC once the obstacle has completely filled the image, when using only one camera. For the experiments where TTC is estimated, it is necessary to issue a stop command when the image from the camera is completely filled, this is achieved by measuring the size of the obstacle in the image and comparing it to the known resolution of the camera. When the size of the object (in pixels) is equal to the camera's resolution, then the image is filled. This can be seen 6 seconds before the end of the manoeuvre in Figure 48.

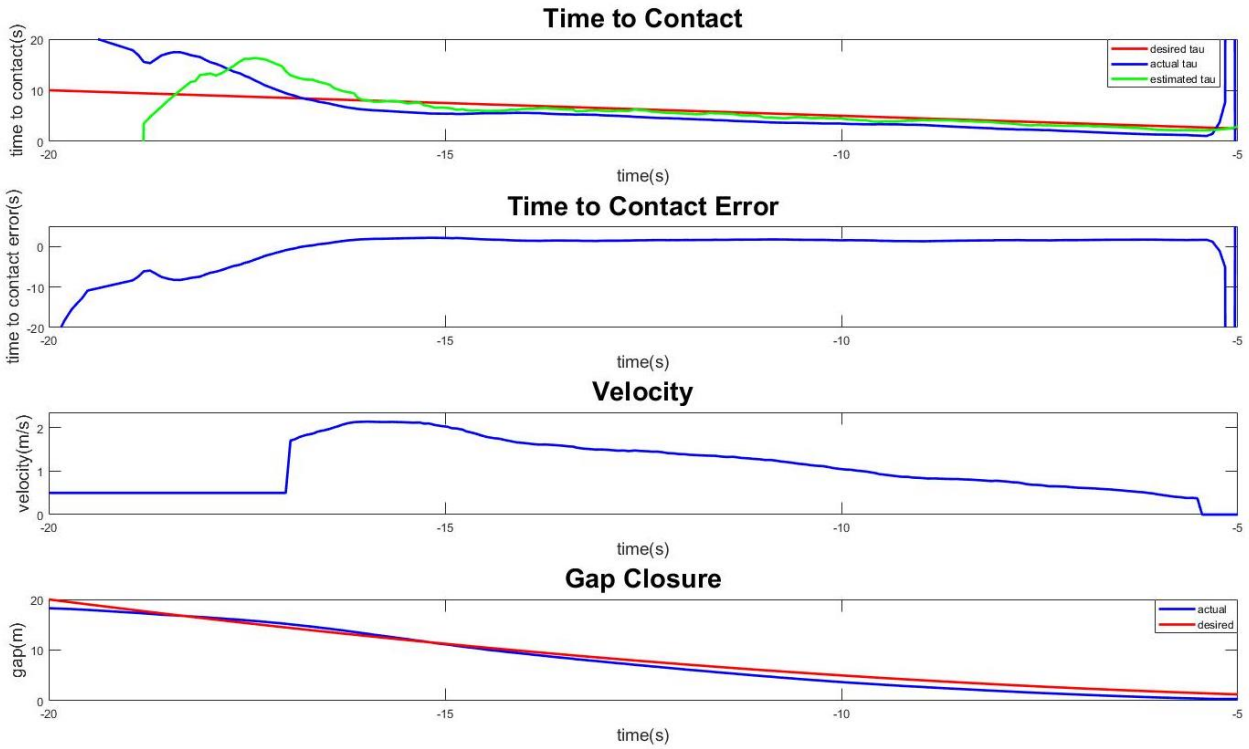


Figure 48. Results from the Simulation Experiment with K = 0.5 and Tau Estimation based on Vision

4.3 Hardware Experiment

Section 4.2 has shown that the linear experiment was successful in simulation but also highlighted some practical issues. Section 4.3 will discuss the results of the experiment which was performed with hardware.

Figure 49 shows the results of the hardware experiment. Like the simulation experiment which used the TTC estimation, there is a small amount of error between the desired TTC and the estimated TTC. Some of this error can be attributed to the noise from the camera images. Again, it is noticeable that there is also some error between the estimated TTC and the actual TTC, as shown in the results of Chapter 3. This error is due to the filtering required to get a smooth TTC estimation, without it, it would not be possible to get a usable TTC estimate.

Figure 49 shows an initial deceleration approximately 3 seconds into the manoeuvre due to the initial desired Tau being higher than the initial estimated Tau. Figure 49 shows that after the initial deceleration phase the robot's Tau is reduced at a constant rate, as is its velocity. It takes approximately 4 seconds for the controller to reduce the error between the desired and actual Tau, this is much longer than that of the simulation experiment but is still usable. This is since K_p(K_p = 0.1) for the hardware experiment had to be lower than that of the simulation experiment (K_p = 1). This was because large sudden changes of velocity cause unreliable Tau estimated from the camera images as the blob detection method was not able to accurately measure the size of the object in the image, whereas no such issue was encountered with the simulated images.

For approximately 10 seconds (-16s to -6s) the robot was under control with minimal Tau error. The velocity appears to be decreasing at a constant rate as desired. At this point there is an apparent gap between the desired and actual gap even though there is minimal error between desired and actual Tau. This is due to the controller aiming to minimise Tau error rather than the gap error. The gap error is therefore caused by

the fact that before this point the robot was correcting some initial Tau error and although it had managed to minimise this error it had done so by decreasing its velocity. This means that the robot is closer to the obstacle but with a lower velocity than it would have been if it had perfectly followed the desired Tau from the very beginning (i.e. no initial deceleration phase).

As discussed previously it was necessary to include an automatic stop when the image of the obstacle filled the entire image. This can be seen approximately 4 seconds before the end of the manoeuvre (time -0.6s). Note that the recorded data stops at approximately -0.6s rather than at the desired 0 seconds. This is due to the automatic stop, after which the data recorded is not useful. For example, the desired Tau would continue to decrease at a constant rate, but the estimate Tau is infinite due to the lack of growth of the obstacle once the robot has stopped.

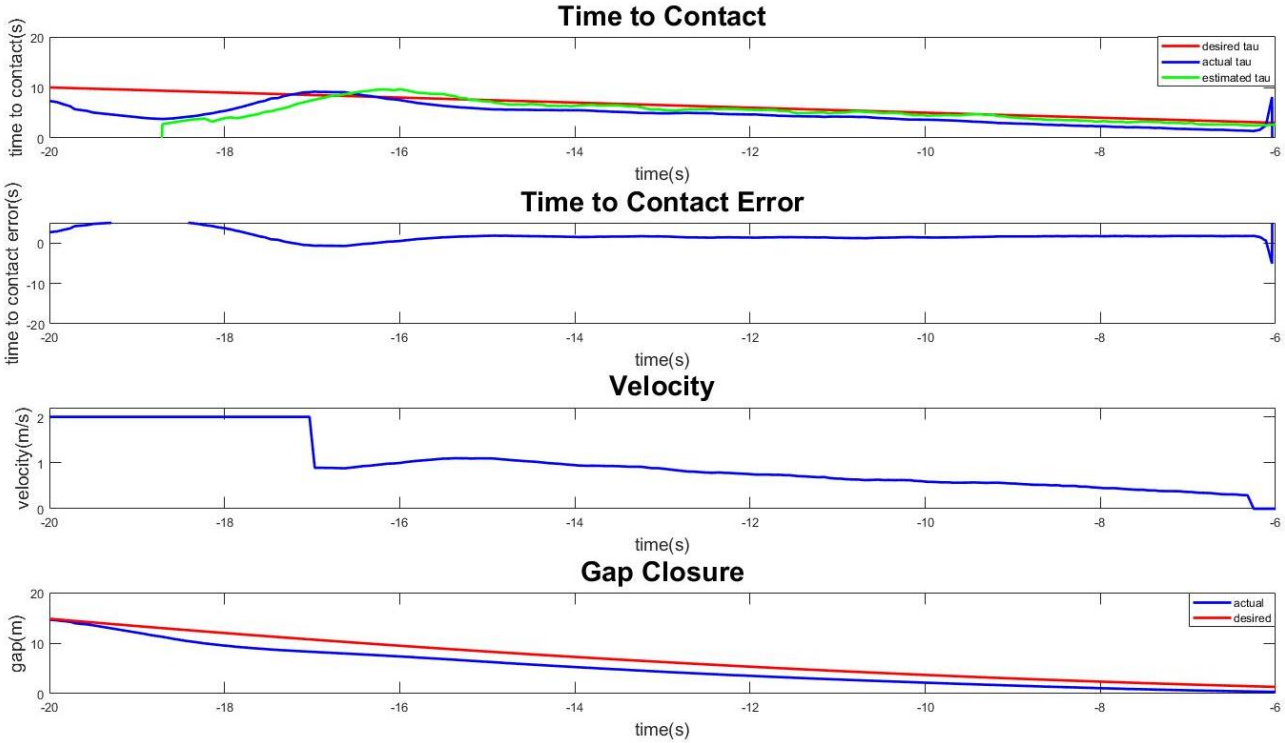


Figure 49. Results Obtained with the Hardware Experiment

4.4 Discussion

This chapter has shown successful experiments for a linear deceleration to a stop. The method described in this chapter could be used for autonomous braking systems in cars or robots. The results have shown that safe, human-like performance is possible when maintaining a constant rate of change of Tau.

The simulation experiments have shown that with a perfect TTC estimation a very smooth manoeuvre can be achieved, therefore the slight noise and error in the TTC estimation method implement in this experiment can be considered a minor weakness. Chapter 6 will discuss potential future work to improve in this area.

Another weakness of this estimation method is the fact that a braking command must be issued when the obstacle fills the image. Humans and animals can navigate even when an object such as a wall fully fills their view if it contains sufficient detail. It is therefore logical that humans and animals are able to obtain TTC information via another method, perhaps by using the texture of the object [65], [99].

4.5 Conclusion

This chapter has introduced the first experiment, a linear approach and deceleration whilst approach an obstacle which models the behaviour of a human driver decelerating and stopping before an obstacle or stop sign. In this experiment the rover uses the Tau estimation method discussed in Chapter 3 and a proportional controller is used to ensure that a constant $\dot{\tau}$ is maintained. Both the hardware and simulated rover were able to maintain a $\dot{\tau}$ with minimal error. This manoeuvre can be considered as showing human-like driving behaviour [27], [34], [35], which may help to ensure their acceptance by human drivers [23], [24].

As discussed in chapter 2 the cutting-edge system is reliant upon a limited set of pre-defined avoidance manoeuvres. This chapter has shown that using Tau-based manoeuvres means that manoeuvres do not have to be pre-defined but can be adjusted automatically by using the Tau strategies such as the constant $\dot{\tau}$ strategy.

CHAPTER 5

LATERAL MANOEUVRE EXPERIMENT

This chapter will discuss the test about using Tau to perform a lateral manoeuvre to avoid an obstacle. This experiment aims to ensure that the vehicle or robot drives around an obstacle and carries on in its original direction. Figure 50 shows how a vehicle or robot would turn left to avoid the obstacle and then right to carry on in its original direction. This manoeuvre ensures that the car still travel along in the x axis to ensure that the car passes the obstacle at the end of the manoeuvre. Therefore, there are two gaps which must be simultaneously closed, the gap in x and the gap in y. The gap in x is defined as the gap between the front of the vehicle and the obstacle, whereas the gap in y is defined as the lateral translation required to safely avoid the obstacle.

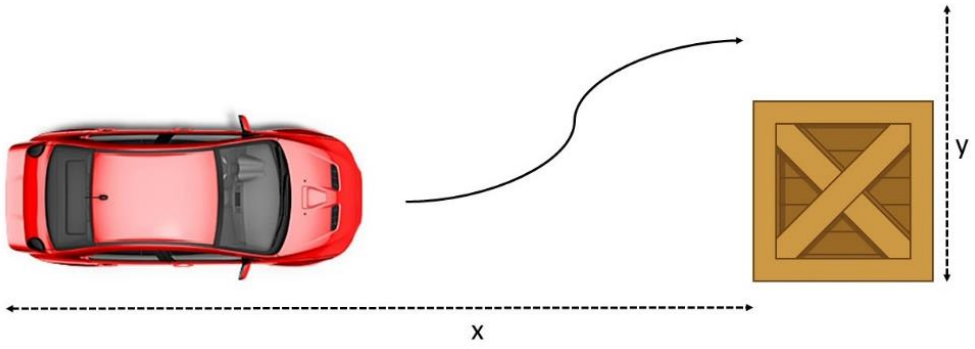


Figure 50. Lateral Manoeuvre

5.1 Generating the Path

The objective of this manoeuvre is to continue moving forwards (along the x axis) whilst moving laterally as to avoid an obstacle. Two Tau guides which are used to

generate the desired path are a constant velocity guide to generate the x position over time and a constant acceleration guide to produce the y position over time. The constant velocity guide was chosen to ensure that the robot accelerates when necessary to ensure that the x gap is closed on time. The constant acceleration guide was chosen due to the curved profile which provides the necessary turn in the manoeuvre.

Figure 51 shows the Tau guides which are used to generate the path as well as the combination of guides necessary to generate the path. Firstly, in the top left of Figure 51 the constant velocity guide is shown. Using this guide ensures that a constant velocity is maintained. Secondly, in the top middle of Figure 51 the constant acceleration guide is shown, this is discussed in Section 2.2. The constant velocity profile is used to close the x gap, this will ensure that a constant linear velocity is achieved. The profile of the constant acceleration guide is useful because there is an acceleration phase at the beginning and a deceleration phase at the end of the manoeuvre, when combining these guides the acceleration guide will be used to close the y gap and will therefore give the necessary curved shape. In the top right of Figure 51 the combination of constant velocity and constant acceleration guides are shown, from this the necessary path shape is created by combining these two guides. This path is then used to create a set of coordinates for the robot to follow around the obstacle. In the bottom left of Figure 51 the orientation over time is shown for the path, it shows how the vehicle or robot first turns in one direction, and then back in the opposite direction to return to its original orientation. In the bottom right of Figure 51 the linear acceleration is shown, the increase of linear acceleration is necessary to keep the velocity in the x axis constant during the turning phase.

Waypoints are a set of coordinates which can be used to designate a path, in this experiment the waypoints will be generated using the guides mentioned previously. Given an x and a y gap a path can be generated as per the top right image of Figure 51. The gaps are measured using the OptiTrack system. Waypoints are generated for

the desired path prior to the beginning of the manoeuvre. The pure pursuit controller receives the waypoints [100], the rover's current pose (x position, y position and rotation) and desired linear velocity and generates the linear and angular velocity commands necessary to follow the desired path. A "look ahead" variable is also used which defines the amount of the desired path which the algorithm can see at any one time. The actual position is tracked using the OptiTrack motion capture system and streamed into MATLAB. The rover stops when it is inside a goal radius (within 10cm of the desired final position), therefore it is unlikely that the rover will stop exactly at the target position. The 10cm goal radius was selected as results from testing showed that this was the smallest value possible before undesired behaviour was introduced. If a smaller goal radius is used, then undesired behaviour such as the circling of the goal are observed.

A system diagram explaining how this process is controlled is shown in Figure 52. The diagram shows how the positional data is external to the system, it is dependent on the OptiTrack system streaming positional data which is required by the pure pursuit algorithm.

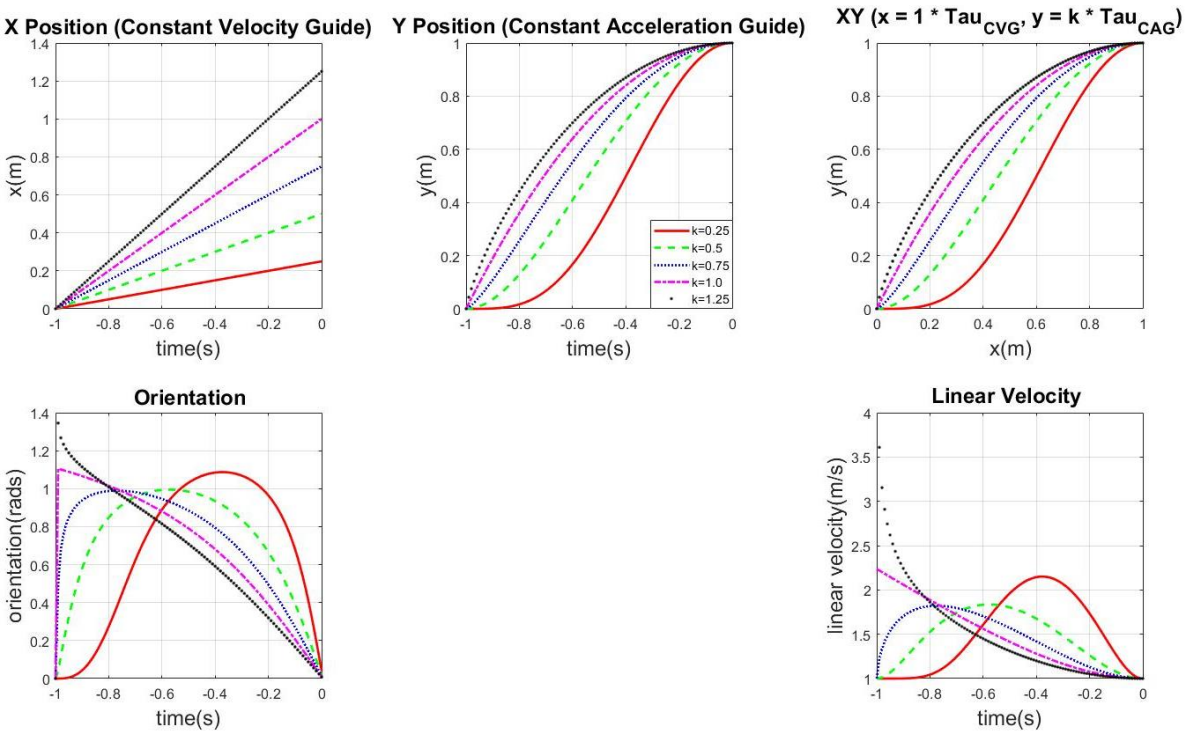


Figure 51. Path Generation for Obstacle Avoidance using Tau Guides

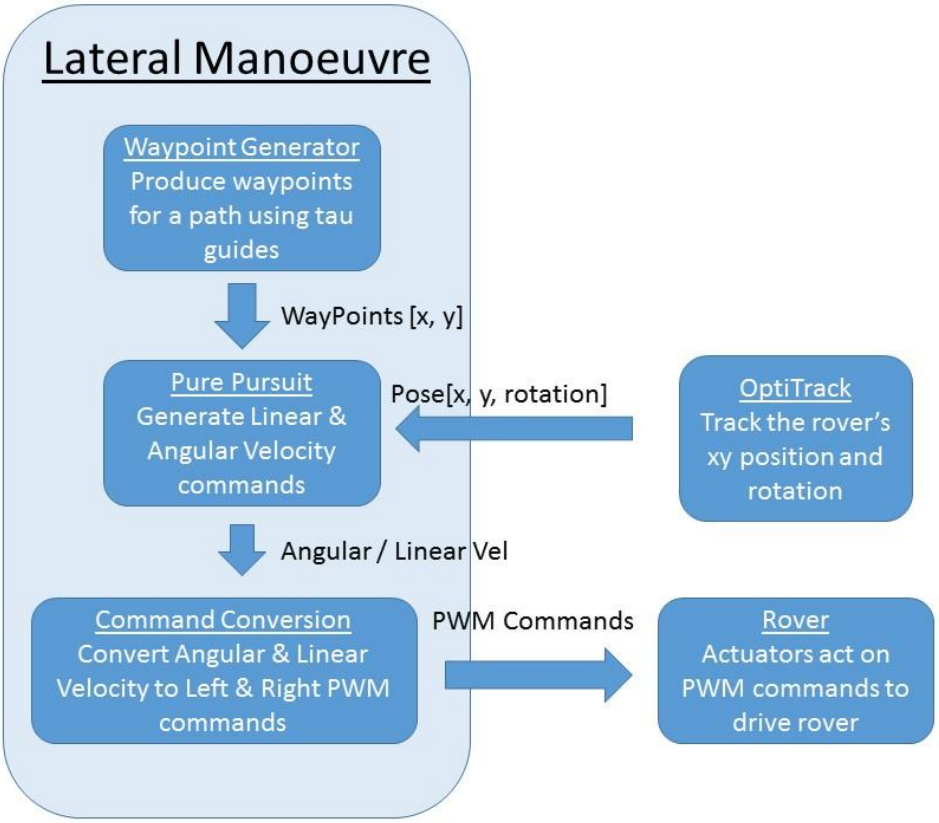


Figure 52. Lateral Manoeuvre System Diagram

5.2 Simulation Results

In this experiment an obstacle was placed 4 metres in front of the model as shown in Figure 53. The model therefore had to move across the y axis by 1 metre to avoid the obstacle at a comfortable margin. The aim is for the model to continue its forwards path across the x axis, so the model needs to move forwards by 5 metres at the same time using the Tau guide combination mentioned earlier. The results show a successful translation, the model successfully followed the path and avoided the obstacle marked in red in Figure 53. Figure 54 shows the error between coordinates over time (in frames), the error is relatively small with a mean error of 0.0294m. An error is always expected when using the pure pursuit algorithm as it does not aim to

move through each point exactly but rather to move as close as possible to the points whilst making progress towards the look ahead point.

Again, this experiment was conducted using the positional data from the simulation model in conjunction with the pure pursuit algorithm which produces the desired linear and angular velocities. There is some error when using the pure pursuit algorithm as tuning produces a trade-off, a look ahead distance that is too short will produce oscillations in the path and a look ahead distance too large will cut corners.

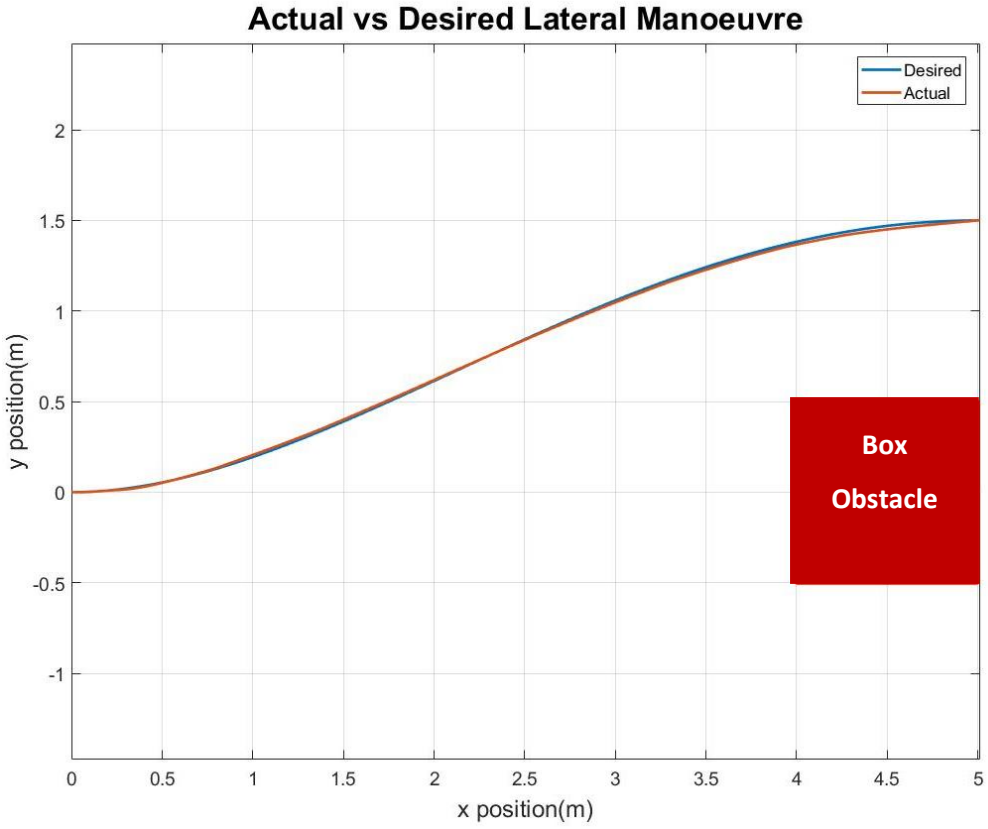


Figure 53. Simulation Lateral Manoeuvre

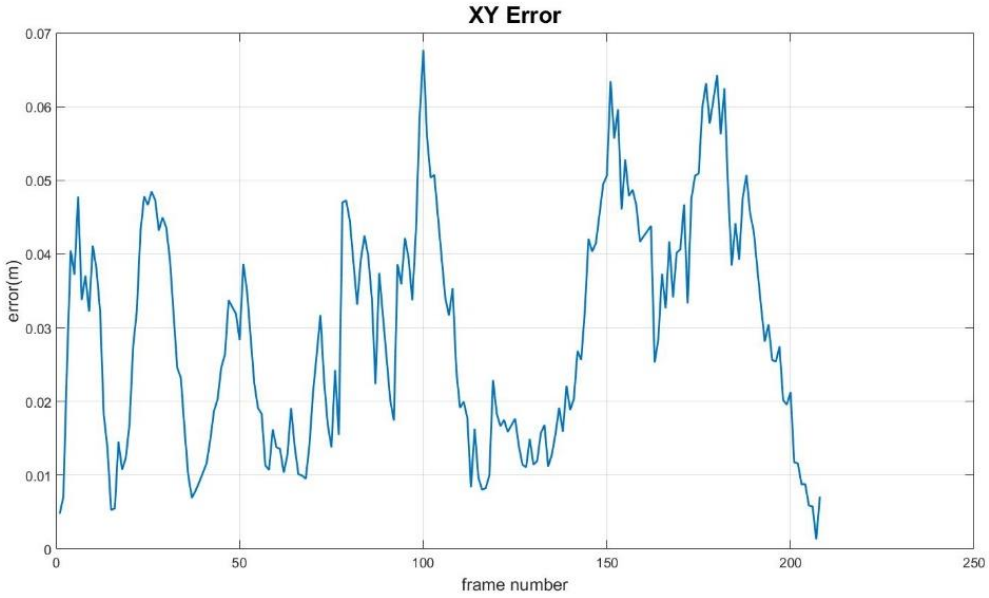


Figure 54. Error of Lateral Manoeuvre

5.3 Hardware Results

Figure 55 shows the result of the lateral avoidance manoeuvre performed on the hardware rover. This experiment is identical to the simulation experiment with the positional truth data being streamed via OptiTrack. Again, there is some error in the path taken due to the nature of the pure pursuit algorithm and its trade-off produced by the look ahead distance. Figure 56 shows the error between the XY coordinates of the desired path and the actual path. There is a maximum error of 0.25m, this error is likely to be caused by the initial delay caused by the rover re-orientating itself which can be seen in Figure 55 between x = -1.5 and x = -1.4.

Similarly, between x = -1.5 and x = -0.8 there is a small gap between the desired route and the actual path taken. The initial cause of this is the re-orientation, however, the robot does not make an immediate effort to close this gap (e.g. by turning right to re-join the path). This again is due to the pure pursuit algorithm which

is not concerned about whether the robot is exactly where it is meant to be but is instead concerned with moving towards some waypoint further ahead. This is a useful feature as it prevents oscillations which may be caused by applying too many corrections.

Between x = 0 and x = 0.4 the line of the actual path becomes less smooth; this is due to a calibration issue with the OptiTrack motion capture lab which could not be corrected in time for the experiment.

The robot stops at approximately x = 0.3 rather than x = 0.4. As mentioned previously, this is due to the pure pursuit algorithm which stops the robot when it is in an acceptable range (0.1m in this instance) of the desired target. Again, this is to prevent the undesired circling of the target.

Appendix B shows a series of images taken of the lateral manoeuvre experiment.

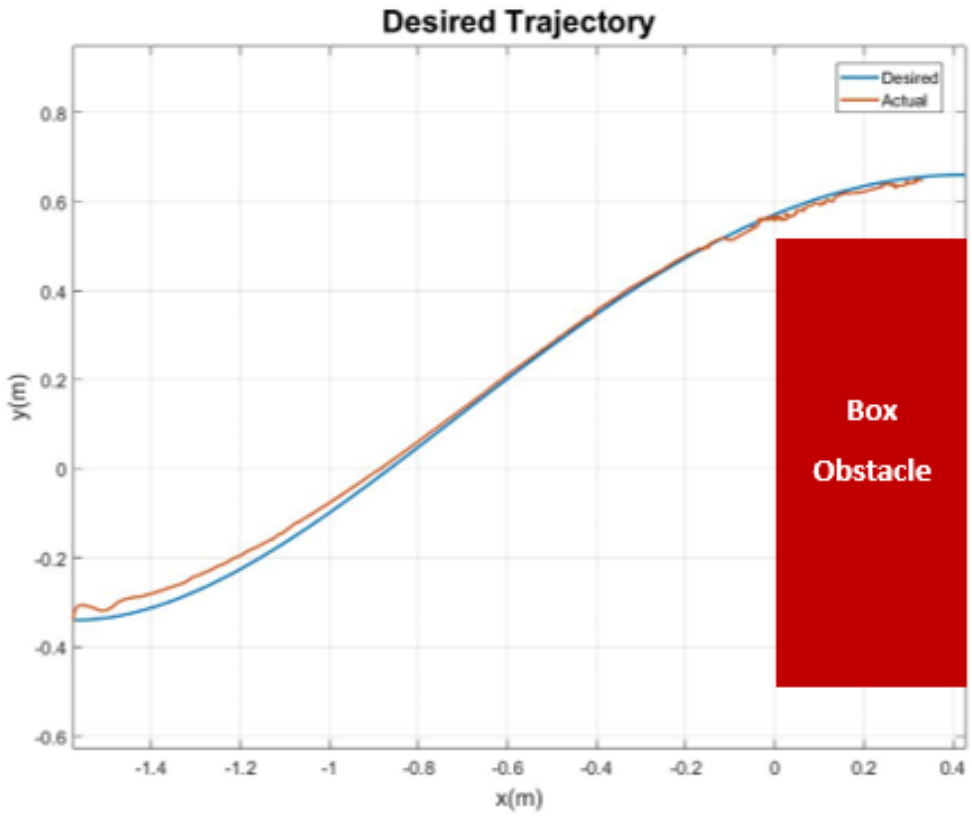


Figure 55. Hardware Lateral Manoeuvre

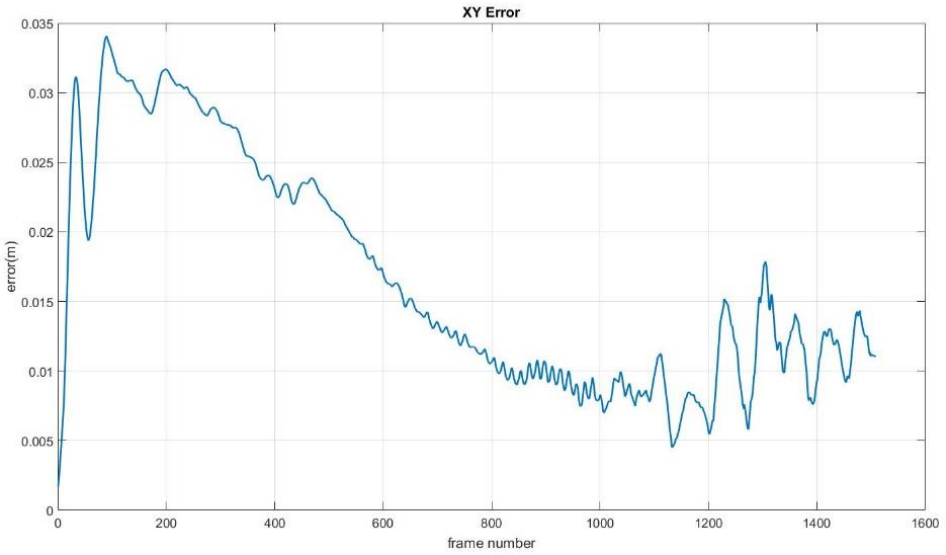


Figure 56. Error of Lateral Manoeuvre

5.4 Discussion

Figure 53 shows that the simulation model has safely passed the obstacle (shown as a red square) and is able to continue in the original direction. Figure 54 shows a small error throughout this manoeuvre, this is likely to have been caused by the nature of the pure pursuit method. This method is unlikely to produce a perfect following of a desired path but with careful tuning of the "look ahead" distance a close match can be produced.

Figure 56 shows a small error throughout the experiment, the largest error appears at the beginning of the manoeuvre. This is caused by the fact that the rover must first re-orientate itself before it may begin to follow the desired path. For the first metre the rover does not quickly close the gap between the actual and desired paths again this is due to the use of the pure pursuit algorithm which generates the velocity commands based on a point further along the desired path based on the "look ahead"

distance. Although the rover does not follow the path perfectly, the general path is followed, and the rover would have avoided the obstacle.

A weakness of this system is the fact that it is dependent upon OptiTrack for positional data. To enable this system to be implemented in more realistic scenarios or on the road, future work is required to implement an on-board perception method. Such a system would need to replace the pure pursuit method as this method is not appropriate when the exact position of the obstacle or robot are not known.

One possible solution could be to close the gap between one side of the obstacle in the image and the side of the image, the x gap in Figure 57. The second gap could be the top of the obstacle in the image and the top of the image, the y gap in Figure 57. The y gap would be closed as the vehicle approaches the object; this would ensure that the linear gap is closed. The x gap would be closed as the vehicle begins to turn away from the obstacle, thus closing the lateral gap. Closing these two gaps simultaneously would create a forward turning manoeuvre like that discussed this chapter, without the need of the pure pursuit algorithm.

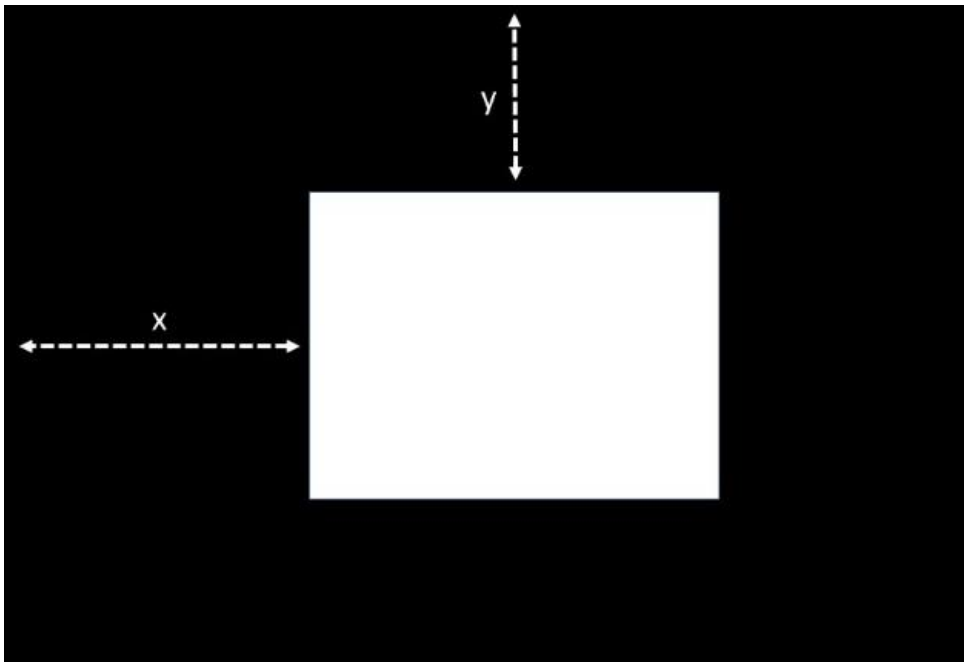


Figure 57. Proposed Visual Gaps

A further improvement to this manoeuvre would be to implement a Tau trigger like that of the linear manoeuvre in Chapter 3. Using this method would allow for the lateral manoeuvre to be triggered only when an obstacle is in the path of vehicle or robot. Currently the manoeuvre is manually triggered, a stationary robot is provided with commands to manoeuvre around the obstacle.

5.5 Conclusion

This chapter has discussed a proposed Tau-based method of obstacle avoidance. An experiment was completed where the constant acceleration guide and the constant velocity guide were combined to produce waypoints to guide a robot around an obstacle. This experiment was successful in that the robot was able to navigate around an obstacle. Suggestions of future work were made, a visual trigger and visual cues for control would make the manoeuvre more practical for a real-world solution.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter presents conclusions to the work discussed within this thesis. It also discusses potential opportunities for further research in the area of bio-inspired navigation for autonomous systems.

6.1 Introduction

This work found within thesis was inspired by the way in which animals and humans navigate cluttered and dynamic environments and how this can be replicated by autonomous vehicles and robots. Tau theory proposes an explanation as to how animals and humans can achieve this using only their senses combined with simple strategies, known as Tau guides. Tau theory proposes that animals achieve useful manoeuvres by closing motion gaps by measuring and reducing their TTC. For example, the driver of a car will close the gap between the front of their car and a stop line when stopping at traffic lights or a stop sign. The driver does not have accurate real-time measurements of this gap, so must therefore use his/her own senses, previous experience, and power of perception to close this gap. This is one of such manoeuvres that the work in this thesis has tried to replicate using robotics.

The three main research components discussed in this thesis are as follows:

1. Implementation of a TTC estimation method (Chapter 3).
2. The linear gap closure strategy of maintaining a constant $\dot{\tau}$ (Chapter 4).
3. Applying Tau guides to the problem of obstacle avoidance (Chapter 5).

6.2 Conclusions of Research

This subsection will provide conclusions of each of the main research components listed in Section 6.1. It will also provide an overall conclusion of the work discussed within this thesis. The results of each experiment are provided in Chapters 3, 4 and 5 of this thesis.

6.2.1 Implementation of a TTC Estimation Method

Chapter 3 discussed the implementation and validation of the TTC estimation method used in the research leading to this thesis. The validation discussed in Chapter 3 shows that a successful measurement was achieved with low error. A significant portion of the error that was found in this validation can be attributed to the filtering techniques which were necessary to ensure that the TTC estimation was usable for the controller. The filtering technique was not changed as it was still able to provide an estimate which achieved the desired manoeuvre whilst using minimal computation power. The TTC estimation results were not perfect, but the estimation was consistent and smooth enough to be used successfully in the controlled stop experiment discussed in Chapter 4.

A weakness of this implementation is the fact that it is dependent on colour isolation for the detection of blobs. A system that was able to detect blobs without isolating colours would allow for the detection of a wider variety of obstacles. Lighting conditions can cause a large variance in the detection of colours. For example, in bright light a colour will seem much lighter in recorded images and it is possible that the red may be seen in reflections of other objects and the floor. Lighting changes often due to the position of the sun in the sky or by the presence of clouds, therefore the detection was often affected by the lighting condition, though the estimation system was only affected is exceptionally poor lighting conditions. This method was used for lab research only and other TTC implementation should be considered for use outside of the controlled laboratory environment.

A commonly asked question is “how does the system tell the difference between a small object and a large object that is far away?”. The answer to this question lies in the growth of the object in the image. Objects that are further away grow more slowly in the image than objects closer to the camera. Equation 12 provides an accurate estimation of Tau regardless of the starting size of the object or the distance away from the camera, providing that the object’s size in the image can be accurately calculated over time.

6.2.2 Linear Gap Closure Strategy

Chapter 2 discussed the constant $\dot{\tau}$ strategy for closing motion gaps, examples of which in nature include bats closing in on prey and birds landing on branches. Chapter 4 is concerned with the closing of a gap between a robot and an obstacle, in this case a red box. This experiment can be thought of as simulating a driver of a car stopping before an obstacle or at traffic lights. This experiment is reliant upon the TTC estimation method discussed in Chapter 3 using images streamed from the camera placed at the front of the robot.

In this experiment the TTC is estimated continuously, when the estimated TTC is lower than a pre-defined safe threshold, a stopping manoeuvre is triggered. The stopping manoeuvre maintains a constant $\dot{\tau}$ strategy. Maintaining a constant $\dot{\tau}$ with a K value less than or equal to 0.5 ensures that as the TTC is reduced to zero, the velocity is also reduced to zero, thus preventing a collision.

Chapter 4 discusses the results of both simulation and hardware experiments using constant $\dot{\tau}$ to ensure that a robot decelerates and stops before it collides with an obstacle. The results show that for both experiments the robot or model decelerated safely to a stop, therefore avoiding collision with the obstacle. The comparison between the experiment using truth data and the experiment using TTC estimations from the camera show estimating TTC from the camera includes some noise and delay and therefore the deceleration is not as smooth. It also shows that it is necessary to stop earlier when using an estimation of TTC instead of the actual TTC value.

This is to be expected when using cameras in this way due to noise from the image and error introduced by moving average filters.

There are several benefits in using these techniques to control a robot or vehicle. Firstly, the equations necessary to estimate TTC are relatively simple when compared to techniques such as machine learning, which is not necessary for the computation of Tau. More simple calculations mean less requirement for computational power, thus providing the opportunity for these programs to be executed on-board smaller processors. Secondly, as discussed in Chapter 1, humans are more likely to accept autonomous vehicles and robots sharing their environment if they behave in a natural way [23], [24]. Tau-based solutions to autonomous systems can generate human-like performance in manoeuvres such as braking. If people are more accepting of autonomous vehicles and robots, then the chances for uptake of such systems are larger and the safety benefits can be extended.

A weakness of this solution relates back to the TTC estimation method used in this project. Since the estimation technique requires an estimation of the expansion of the obstacle in the image, it ceases to provide a TTC estimation once the obstacle completely fills the camera image if only a single camera is available. Therefore, it is necessary to issue a stop command as soon as this occurs. The results in Chapter 4 show that even with this limitation it is possible to safely and smoothly control a robot to a stop, but it is necessary to stop earlier than desired, depending on the size of the obstacle and the properties of the camera.

6.2.3 Tau Guides for Obstacle Avoidance

Chapter 5 discussed the utilisation of both a constant velocity Tau guide and a constant acceleration Tau guide to generate natural paths to avoid an obstacle. The robot follows such paths using a pure pursuit algorithm which requires accurate data of the robot's location and orientation. Results for both simulation and hardware experiment were discussed.

Two experiments were performed successfully, firstly a simulation experiment using data from the simulation environment was performed, and secondly a hardware experiment was performed using tracking data from the OptiTrack system. The results of these experiments showed that with accurate positional data the path can be followed accurately, but not perfectly due to the nature of the pure pursuit algorithm.

Although the experiments were successful, this work has highlighted several weaknesses which must be addressed before it can operate outside of the laboratory environment. Firstly, the dependence on external tracking data must be removed and to do this the pure pursuit method of control must be replaced. Potential options to remove this weakness will be discussed in Section 6.3. Secondly, in the current implementation of this experiment, the Tau guides are used only to generate a path for the robot to follow using the pure pursuit algorithm. This is not a natural method of guidance, to implement a true Tau guidance strategy the motion gaps should be closed using TTC. A potential solution to this problem is discussed in Section 6.3.

6.2.4 Overall Conclusions

The overall aim of this project was to develop autonomous guidance strategies inspired by nature. Chapter 3 has discussed a successful Tau estimation system based on optical expansion which have been used to control a linear manoeuvre discussed in Chapter 4. Chapters 4 and 5 have discussed the results of experiments for both a linear and lateral manoeuvre, both of which were successful. It is hoped that by implementing autonomous guidance methods that more closely replicate human behaviour that they will be more accepted, and the uptake of autonomous systems will increase. Overall, the results of the experiments discussed in this thesis suggest that Tau-based guidance methods can be used safely to replicate natural guidance manoeuvres for autonomous systems. This chapter has also discussed some suggestions for future work which would improve the current system and make it more practical in the real-world.

Chapter 2 identified the CSAIL system as being the cutting-edge in perception and obstacle avoidance. It also identified two key limitations of the system. Firstly, the system does not detect obstacles in the first 10 metres of its fly nor for the 10 metres after a sharp turn. Secondly, the system is reliant upon a limited number of pre-defined avoidance manoeuvres, when the system finds itself in a scenario for which there is no defined avoidance strategy it crashes. This thesis has demonstrated how Tau-based systems improve upon these two limitations. Chapter 2 showed how by using Tau as a method of perception obstacles can be constantly sensed at all points of the manoeuvre. Chapters 3 and 4 demonstrated how Tau-based guidance strategies do not have to be pre-defined but instead can be automatically adapted for individual situations.

6.3 Future Work

Whilst this thesis presents several successful experiments, there is still much work required to before Tau-based guidance systems can be safely used outside of the laboratory environment. Chapter 6 has summarised a few limitations with the research presented within this thesis. This subsection aims to provide guidance and inspiration for further work in autonomous guidance.

6.3.1 TTC Estimation Improvement

Whilst the TTC estimation method is dependent upon colour extraction, it is unlikely that the current system will be usable outside of the laboratory setting. To ensure that the TTC estimation method is usable outside of such a setting the following improvements must be developed:

1. Obstacles in the image should be detected without the colour range method ensuring that obstacles are consistently and accurately detected independent of lighting conditions. Optic flow based methods of calculating Tau show promise in this area, however, they bring limitations of their own as discussed in chapter 1 [7], [59], [61].

2. A method of estimating TTC when the obstacle has filled the image must be developed to ensure that the autonomous system can operate in more scenarios. For example, when large obstacles such as buildings fill the entire image. A common approach to this problem is to estimate TTC using optical flow Chapter 2 reviewed a number of examples of such systems [58], [65], [73], [99], [101]. Again, the systems reviewed had their own limitations such as noisy estimation for certain textures and a dependency on constant brightness.

6.3.2 Lateral Manoeuvre Improvement

As discussed in chapter 5 existing solutions such as the pure pursuit algorithm have significant limitations when used for the purpose of Tau-based lateral manoeuvres. The lateral manoeuvre experiment discussed in Chapter 5 requires the following further work to ensure that it can execute autonomously:

1. Instead of simply following a path, the system should identify and close motion gaps. These may take the form of gaps in the image, for example, Section 5.4 discusses the possibility of using the gaps between the edge of the obstacle in an image and the edge of the image. The literature review performed in the development of this thesis did not discover any Tau-based solutions to this problem.
2. On-board perception is required to estimate the closure of such gaps.
3. A control system must be developed to allow for the lateral manoeuvre to be completed by controlling TTC instead of using the pure pursuit algorithm discussed in Chapter 5.

REFERENCES

[1] F. Kendoul, “Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems,” *J. F. Robot.*, vol. 29, no. 2, pp. 315–378, 2012, doi: 10.1002/rob.20414.

[2] S. Singh, J. Mishler, H. Cover, and B. Hamner, “Perception for Safe Autonomous Helicopter Flight and Landing,” in *AHS 72nd Annual Forum*, 2016, pp. 1–8.

[3] S. E. D. N. K, “Vision-Based 3D Navigation for an Autonomous Helicopter,” *Most*, no. May, 2006.

[4] A. J. Barry, “High-Speed Autonomous Obstacle Avoidance with Pushbroom Stereo,” Massachusetts Institute of Technology, 2016.

[5] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, “PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision,” *Auton. Robots*, vol. 33, no. 1–2, pp. 21–39, 2012, doi: 10.1007/s10514-012-9281-4.

[6] M. S. Whalley *et al.*, “Autonomous Black Hawk in Flight: Obstacle Field Navigation and Landing-site Selection on the RASCAL JUH-60A,” *J. F. Robot.*, vol. 31, no. 4, pp. 591–616, 2014, doi: 10.1002/rob.

[7] F. Kendoul and B. Ahmed, “Bio-inspired TauPilot for automated aerial 4D docking and landing of Unmanned Aircraft Systems,” in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 480–487, doi: 10.1109/IROS.2012.6385586.

[8] T. Mori and S. Scherer, “First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles,” in

Proceedings - IEEE International Conference on Robotics and Automation, 2013, pp. 1750–1757, doi: 10.1109/ICRA.2013.6630807.

[9] I. Sa, H. He, V. Huynh, and P. Corke, "Monocular vision based autonomous navigation for a cost-effective MAV in GPS-denied environments," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing, AIM 2013*, 2013, pp. 1355–1360, doi: 10.1109/AIM.2013.6584283.

[10] R. Hussain, "Autonomous Cars : Research Results , Issues , and Future Challenges," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2019, doi: 10.1109/COMST.2018.2869360.

[11] J. Steward, "Tesla's AutoPilot was Involved in Another Deadly Crash," *Wired*, 2018. https://www.wired.com/story/tesla-autopilot-self-driving-crash-california/ (accessed May 30, 2018).

[12] BBC News, "Tesla Model 3: Autopilot engaged during fatal crash," 2019. .

[13] B. Templeton, "Tesla Autopilot Repeats Fatal Crash; Do They Learn From Past Mistakes?," *Forbes*, 2019. .

[14] S. Jack, "Why Tesla's Autopilot Can't See a Stopped Firetruck," *Wired*, 2018. https://www.wired.com/story/tesla-autopilot-why-crash-radar/ (accessed Feb. 03, 2020).

[15] D. Robineau, "Reported road casualties in Great Britain : 2017 annual report," 2018.

[16] J. J. Rolison, S. Regev, S. Moutari, and A. Feeney, "What are the factors that contribute to road accidents ? An assessment of law enforcement views , ordinary drivers ' opinions , and road accident records," *Accid. Anal. Prev.*, vol. 115, no. February, pp. 11–24, 2018, doi: 10.1016/j.aap.2018.02.025.

[17] S. Kato, E. Takeuchi, Y. Ishiguro, Ninomiya, Yoshiki, K. Takeda, and T. Hamada, “An Open Approach to Autonomous Vehicles,” *IEEE Micro*, pp. 60–68, 2015.

[18] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli, “Real-Time Obstacle Detection using Stereo Vision for Autonomous Ground Vehicles : A Survey,” *17th Int. IEEE Conf. Intell. Transp. Syst.*, no. 1, pp. 873–878, 2014, doi: 10.1109/ITSC.2014.6957799.

[19] H. Abraham, C. Lee, B. Mehler, and B. Reimer, “Autonomous Vehicles and Alternatives to Driving : Trust , Preferences , and Effects of Age,” in *Transportation Research Board 96th Annual Meeting*, 2017, no. January.

[20] J. Gorzelany, “Volvo Will Accept Liability for Its Self-Driving Cars,” *Forbes*, 2015. .

[21] A. Frison, P. Wintersberger, and A. Riener, “First Person Trolley Problem : Evaluation of Drivers ’ Ethical Decisions in a Driving Simulator,” in *Adjunct Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2016.

[22] S. Nyholm and J. Smids, “The Ethics of Accident-Algorithms for Self-Driving Cars : an Applied Trolley Problem ?,” *Ethical Theory Moral Pract.*, pp. 1275–1289, 2016, doi: 10.1007/s10677-016-9745-2.

[23] E. Broadbent, B. M. S. Member, L. Jago, M. Juergens, and O. Mazharullah, “Human reactions to good and bad robots,” in *Proceedings of the 2007 IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 3703–3708.

[24] P. D. Van Greunen, P. Elizabeth, and S. Africa, “User Experience for Social Human-Robot Interactions,” *2019 Amity Int. Conf. Artif. Intell.*, pp. 32–36, 2019.

[25] G. D. Padfield, “The tau of flight control,” *Aeronaut. J.*, vol. 115, no. 1171, pp. 521–556, 2011.

[26] M. Jump, “Developing Guidelines for Pilot Vision,” University of Liverpool, 2007.

[27] D. N. Lee, “Guiding Movement by Coupling Taus.,” *Ecol. Psychol.*, vol. 10, no. 3/4, pp. 221–251, 1998, doi: 10.1207/s15326969eco103&4_4.

[28] D. Lee, “General Tau Theory: evolution to date,” *Perception*, vol. 38, no. 6, pp. 854–857, 2009, doi: 10.1068/ldmk-lee.

[29] D. N. Lee, “Tau in action in development,” *Action, Percept. Cogn. Learn. Dev.*, p. 58, 2004, [Online]. Available: papers://4dd355a6-509d-432d-a6d3-eb12a1988728/Paper/p50.

[30] D. N. Lee, P. E. Reddish, and D. T. Rand, “Aerial docking by hummingbirds,” *Naturwissenschaften*, vol. 78, no. 11, pp. 526–527, 1991, doi: 10.1007/BF01131406.

[31] D. N. Lee, “How Movement is Guided,” 2011. doi: 10.1017/CBO9781107415324.004.

[32] M. Jump and G. D. Padfield, “Investigation of the Flare Maneuver Using Optical Tau,” *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5. pp. 1189–1200, 2006, doi: 10.2514/1.20012.

[33] G. D. Padfield, D. N. Lee, and R. Bradley, “How Do Helicopter Pilots Know When to Stop, Turn or Pull Up?(Developing Guidelines for Vision Aids),” *J. Am. Helicopter Soc.*, vol. 48, no. 2, pp. 108–119, 2003, doi: 10.4050/JAHS.48.108.

[34] B. Hopkins, A. Churchill, S. Vogt, and L. Rönnqvist, “Braking reaching movements: a test of the constant tau-dot strategy under different viewing

conditions.," *J. Mot. Behav.*, vol. 36, no. 1, pp. 3–12, 2004, doi: 10.3200/JMBR.36.1.3-12.

[35] E. H. Yilmaz and W. H. Warren, "Visual control of braking: A test of the τ̇ hypothesis.," *J. Exp. Psychol. Hum. Percept. Perform.*, vol. 21, no. 5, pp. 996–1014, 1995, [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/0096-1523.21.5.996%5Cnpapers3://publication/doi/10.1037//0096-1523.21.5.996.

[36] D. N. Lee, "Visual control of locomotion," pp. 224–230, 1977.

[37] A. J. Sanchez-Garcia, H. V. Rios-Figueroa, A. Marin-Hernandez, and G. Contreras-Vega, "Decision making for obstacle avoidance in autonomous mobile robots by time to contact and optical flow," *25th Int. Conf. Electron. Commun. Comput. CONIELECOMP 2015*, pp. 130–134, 2015, doi: 10.1109/CONIELECOMP.2015.7086939.

[38] G. E. Smith, C. J. Baker, and G. Li, "Echoic flow for radar target interception," *2013 Int. Conf. Radar - Beyond Orthodox. New Paradig. Radar, RADAR 2013*, pp. 520–525, 2013, doi: 10.1109/RADAR.2013.6652042.

[39] Z. Zhang, S. Zhang, P. Xie, and O. Ma, "Bioinspired 4D trajectory generation for a UAS rapid point-to-point movement," *J. Bionic Eng.*, vol. 11, no. 1, pp. 72–81, 2014, doi: 10.1016/S1672-6529(14)60021-4.

[40] Y. Kaneta, Y. Hagisaka, and K. Ito, "Determination of time to contact and application to timing control of mobile robot," in *2010 IEEE International Conference on Robotics and Biomimetics*, 2010, vol. 2, no. 1, pp. 161–166, doi: 10.1109/ROBIO.2010.5723320.

[41] G. D. Padfield, L. H. Lu, and M. Jump, "Tau Guidance in Boundary-

Avoidance Tracking: New Perspectives on Pilot-Induced Oscillations," *J. Guid. Control Dyn.*, vol. 35, no. 1, pp. 80–92, 2012, doi: Doi 10.2514/1.54065.

[42] Optitrack, "Optitrack - Motion Capture Systems," 2018. https://optitrack.com/ (accessed May 23, 2018).

[43] OpenCV, "Motion Analysis and Object Tracking," *OpenCV Docs*, 2019. https://docs.opencv.org/2.4/modules/video/doc/motion_analysis_and_object_tracking.html (accessed Feb. 02, 2020).

[44] A. J. Barry, H. Oleynikova, D. Honegger, M. Pollefeys, and R. Tedrake, "Fast Onboard Stereo Vision for UAVs," in *IROS Workshop*, 2015, p. 7, [Online]. Available: http://groups.csail.mit.edu/robotics-center/public_papers/Barry15a.pdf.

[45] A. J. Barry and R. Tedrake, "Pushbroom Stereo for High-Speed Navigation in Cluttered Environments," in *3rd Workshop on Robots in Clutter: Perception and Interaction*, 2014, pp. 2–8, doi: 10.1109/ICRA.2015.7139617.

[46] T. Merz and F. Kendoul, "Dependable low-altitude obstacle avoidance for robotic helicopters operating in rural areas," *J. F. Robot.*, vol. 30, no. 3, pp. 439–471, 2013, doi: 10.1002/rob.21455.

[47] B. Mettler, Z. Kong, C. Goerzen, and M. Whalley, "Benchmarking of obstacle field navigation algorithms for autonomous helicopters," in *Proceedings of the 66th Annual Forum of the American Helicopter Society*, 2010, pp. 1–18.

[48] M. Whalley, M. Takahashi, G. J. Schulein, and C. Goerzen, "Field-testing of a helicopter UAV obstacle field navigation and landing system," in *65th Annual Forum of the American Helicopter Society*, 2009, [Online].

Available: http://armyscienceconference.com/manuscripts/E/EP-008.pdf.

[49] J. Michels, A. Saxena, and A. Y. Ng, “High speed obstacle avoidance using monocular vision and reinforcement learning,” in *Proceedings of the 22nd international conference on Machine learning - ICML ’05*, 2005, vol. 3, no. Suppl 1, pp. 593–600, doi: 10.1145/1102351.1102426.

[50] S. Mitsch, K. Ghorbal, and A. Platzer, “On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles,” *Robot. Sci. Syst.*, 2013, [Online]. Available: http://roboticsproceedings.org/rss09/p14.pdf%5Cnhttp://www.roboticsproceedings.org/rss09/p14.pdf.

[51] M. D. Takahashi *et al.*, “Development and Flight Testing of a Flight Control Law for Autonomous Operations Research on the RASCAL JUH-60A,” *J. Am. Helicopter Soc.*, vol. 59, pp. 1–13, 2014, doi: 10.4050/JAHS.59.032007.

[52] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma, “Flying Fast and Low Among Obstacles: Methodology and Experiments,” *Int. J. Rob. Res.*, vol. 27, no. 5, pp. 549–574, 2008, doi: 10.1177/0278364908090949.

[53] L. Ren, W. Wang, and Z. Du, “Intelligent obstacle avoidance control strategy for wheeled mobile robot,” in *Proceedings of 2012 IEE International Conference on Mechatronics and Automation*, 2012, pp. 1732–1737, [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5334116.

[54] L. Ren, W. Wang, and Z. Du, “Intelligent Obstacle Avoidance Control Strategy for Wheeled Mobile Robot,” in *ICROS-SICE INternational Joint Conference 2009*, 2009, pp. 3199–3204.

[55] M. Jump, F. Science, and B. Hill, “Developing Guidelines for an Approach and Landing Sky,” no. August, pp. 1–18, 2005.

[56] Z. Yang, Z. Fang, and P. Li, “Bio-inspired Collision-free 4D Trajectory Generation for UAVs Using Tau Strategy,” *J. Bionic Eng.*, vol. 13, no. 1, pp. 84–97, 2016, doi: 10.1016/S1672-6529(14)60162-1.

[57] Z. Zhang, P. Xie, and O. Ma, “Bio-inspired Trajectory Generation for UAV Perching Movement Based on Tau Theory,” *Int. J. Adv. Robot. Syst.*, p. 1, 2014, doi: 10.5772/58898.

[58] X. Clady *et al.*, “Asynchronous visual event-based time-to-contact.,” *Front. Neurosci.*, vol. 8, no. February, p. 9, 2014, doi: 10.3389/fnins.2014.00009.

[59] K. Souhila, O. Djekoune, D. Djebrouni, and D. Meriche, “On the use of optical flow in robot navigation,” in *IEEE International Conference on Signal Processing and Communications*, 2007, no. November, pp. 24–27.

[60] G. de Croon, D. Izzo, and G. Schiavone, “Time-to-Contact Estimation in Landing Scenarios Using Feature Scales,” *Acta Futur.*, vol. 5, pp. 73–82, 2012, doi: 10.2420/AF05.2012.73.

[61] D. Izzo and G. De Croon, “Landing with Time-to-Contact and Ventral Optic Flow Estimates,” *J. Guid. Control. Dyn.*, vol. 35, no. 4, pp. 1362–1367, 2012, doi: 10.2514/1.56598.

[62] G. Alenyà, A. Nègre, and J. L. Crowley, “Time to contact for obstacle avoidance,” *Eur. Conf. Mob. Robot.*, pp. 1–6, 2009.

[63] M. T. Alkowatly, V. M. Becerra, and W. Holderbaum, “Estimation of Visual Motion Parameters Used for Bio-inspired Navigation,” *J. Image Graph.*, vol. 1, no. 3, pp. 120–124, 2013, doi: 10.12720/joig.1.3.120-124.

[64] G. De Croon, D. Alazard, and D. Izzo, “Controlling spacecraft landings with constantly and exponentially decreasing time-to-contact,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 2, pp. 1241–1252, 2015, doi: 10.1109/TAES.2014.130135.

[65] E. Rondon, I. Fantoni-Coichot, A. Sanchez, and G. Sanahuja, “Optical flow-based controller for reactive and relative navigation dedicated to a four rotor rotorcraft,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, 2009, pp. 684–689, doi: 10.1109/IROS.2009.5354483.

[66] S. Tijmons, G. de Croon, B. Remes, C. De Wagter, and M. Mulder, “Obstacle Avoidance Strategy using Onboard Stereo Vision on a Flapping Wing MAV,” *IEEE Trans. Robot.*, no. April, 2016, [Online]. Available: http://arxiv.org/abs/1604.00833.

[67] M. S. Guzel and R. Bicker, “Optical flow based system design for mobile robots,” *Robot. Autom. Mechatronics (RAM), 2010 IEEE Conf.*, pp. 545–550, 2010, doi: 10.1109/RAMECH.2010.5513134.

[68] C. Braillon, J. L. Crowley, and C. Laugier, “Real-time Time-To-Collision from variation of Intrinsic Scale,” in *Proceedings of the International Symposium on Experimental Robotics*, 2007, pp. 1–10.

[69] F. Kendoul, “Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems,” *Int. J. Rob. Res.*, vol. 33, no. 2, pp. 237–267, 2013, doi: 10.1177/0278364913509496.

[70] P. Nebot and E. Cervera, “Cooperative navigation using the optical flow and time-to-contact techniques,” *2008 10th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2008*, no. December, pp. 1736–1741, 2008, doi: 10.1109/ICARCV.2008.4795789.

[71] D. Howard and Kendoul Farid, “Towards Evolved Time to Contact Neurocontrollers for Quadcopters,” in *Artificial Life and Computational Intelligence*, no. July, 2015, pp. 336–347.

[72] B. K. B. Horn and B. G. Schunck, “Determining Optical Flow,” *Artif. Intell. Memo 572*, vol. 319, pp. 185–203, 1980, doi: 10.1016/0004-3702(81)90024-2.

[73] B. K. P. Horn, Y. F. Y. Fang, and I. Masaki, “Time to Contact Relative to a Planar Surface,” in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 68–74, doi: 10.1109/IVS.2007.4290093.

[74] T. Camus, “Calculating Time-to-Contact Using Real-Time Quantized Optical Flow,” 1995.

[75] W. A. Arokiasami and T. K. Chen, “Impact of the Length of Optical Flow Vectors in Estimating Time-to-Contact an Obstacle,” in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, 2015, vol. 2, pp. 201–213, doi: 10.1007/978-3-319-13356-0.

[76] H. Chao, Y. Gu, and M. Napolitano, “A survey of optical flow techniques for robotics navigation applications,” *J. Intell. Robot. Syst. Theory Appl.*, vol. 73, no. 1–4, pp. 361–372, 2014, doi: 10.1007/s10846-013-9923-6.

[77] D. K. Liyanage and M. U. S. Perera, “Optical flow based obstacle avoidance for the visually impaired,” *2012 IEEE Business, Eng. Ind. Appl. Colloq.*, pp. 284–289, 2012, doi: 10.1109/BEIAC.2012.6226068.

[78] V. Bruce, P. . Green, and M. . Georgeson, “Visual Perception: Physiology, Psychology and Ecology,” *Hove Psychol. Press*, 1996.

[79] I. . George, “Theories of Visual Perception,” *John Wiley Sons*, 1989.

[80] J. Engel, J. Sturm, and D. Cremers, “Camera-based navigation of a low-cost quadrocopter,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2815–2821, doi: 10.1109/IROS.2012.6385458.

[81] Dube. Parijat and Z. Sura, “Impact of System Resources on Performance of Deep Neural Network,” in *2018 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2018, no. 2, pp. 125–127, doi: 10.1109/ISPASS.2018.00025.

[82] Dr Robot Inc, “Dr Robot Inc.: WiFi 802.11 robot, Network-based Robot, robotic, robot kit, humanoid robot, OEM solution,” 2017. http://jaguar.drrobot.com/specification_4x4w.asp (accessed Jan. 04, 2017).

[83] M. Zohaib, M. Pasha, R. A. Riaz, J. Nadeem, M. Ilahi, and R. D. Khan, “Control Strategies for Mobile Robot With Obstacle Avoidance,” *CoRR*, vol. abs/1306.1, pp. 1027–1036, 2013, [Online]. Available: http://arxiv.org/abs/1306.1144.

[84] M. T. Alkowatly, V. M. Becerra, and W. Holderbaum, “Bioinspired Autonomous Visual Vertical Control of a Quadrotor Unmanned Aerial Vehicle,” *J. Guid. Control. Dyn.*, vol. 38, no. 2, pp. 249–262, 2015, doi: 10.2514/1.G000634.

[85] OSRF Foundation, “Gazebo,” 2014. http://gazebosim.org/ (accessed Jan. 04, 2017).

[86] MathWorks, “Simulink,” 2019. .

[87] Itseez, “OpenCV | OpenCV,” 2017. http://opencv.org/ (accessed Jan. 04, 2017).

[88] DARPA, “DARPA’s legacy: Open source simulation for robotics development and testing | Robohub.” http://robohub.org/darpas-legacy-open-source-simulation-for-robotics-development-and-testing/ (accessed Jan. 04, 2017).

[89] E. Ackerman, “DARPA Awards Simulation Software Contract to Open Source Robotics Foundation - IEEE Spectrum,” 2012.

http://spectrum.ieee.org/automaton/robotics/robotics-software/darpa-robotics-challenge-simulation-software-open-source-robotics-foundation (accessed Jan. 04, 2017).

[90] Open Source Robotics Foundation, “SDFormat,” *Open Source Robotics Foundation*, 2019. http://sdformat.org/spec (accessed Mar. 28, 2020).

[91] Open Source Robotics Foundation, “Tutorial: Using a URDF in Gazebo,” *Gazebo*, 2014. http://gazebosim.org/tutorials?tut=ros_urdf (accessed Mar. 28, 2020).

[92] MathWorks, “Matlab,” 2019. .

[93] National Instruments, “PID Theory Explained,” 2019. .

[94] OSRF Foundation, “ROS.org | Powering the world’s robots.” http://www.ros.org/ (accessed Jan. 04, 2017).

[95] M. Quigley *et al.*, “ROS: an open-source Robot Operating System,” in *ICRA Workshop on Open Source Software*, 2009, vol. 3, pp. 1–6, doi: http://www.willowgarage.com/papers/ros-open-source-robot-operating-system.

[96] OSRF Foundation, “Robotics Projects,” 2017. http://www.osrfoundation.org/osrf-projects/ (accessed Jan. 04, 2017).

[97] D. Kohanbash, “Drive Kinematics: Skid Steer & Mecanum (ROS Twist included),” *robots for roboticists*, 2016. .

[98] L. Lu, M. Jump, and M. Jones, “Tau Coupling Investigation Using Positive Wavelet Analysis,” *J. Guid. Control. Dyn.*, vol. 36, no. 4, pp. 920–934, 2013, doi: 10.2514/1.60015.

[99] G. D. Illeperuma and U. J. Sonnadara, “An autonomous robot navigation system based on optical flow,” *2011 6th Int. Conf. Ind. Inf. Syst.*, no. March,

pp. 489–492, 2011, doi: 10.1109/ICIINFS.2011.6038119.

[100] R. C. Coulter, “Implementation of the Pure Pursuit Path Tracking Algorithm,” 1992. doi: CMU-RI-TR-92-01.

[101] J. C. Zufferey and D. Floreano, “Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2005, no. April, pp. 2594–2599, 2005, doi: 10.1109/ROBOT.2005.1570504.

APPENDIX A

This appendix provides the proof to Equation 18.

$$A = \frac{C}{x^2} = Cx^{-2}$$

$$\dot{A} = \Theta x^{-1} + C\frac{d}{\Delta t}(x^{-1})$$

$$\dot{A} \sim \frac{A(t + \Delta t) - A(t)}{\Delta t}$$

$$= \left(\frac{C}{x_{t+\Delta t}^2} - \frac{C}{x_t^2}\right)\frac{1}{\Delta t}$$

Since $x_{t+\Delta t} \sim x_t + \dot{x}\Delta t$

$$= \frac{1}{\Delta t}\left\{\frac{C}{(x_t + \dot{x}\Delta t)^2} - \frac{C}{x_t^2}\right\}$$

$$= \frac{C}{\Delta t}\left\{\frac{x_t^2 - \left(x_t + \dot{x\Delta t}\right)^2}{(x_t^2 + \dot{x})^2\, x_t^2}\right\}$$

$$= \frac{C}{\Delta t}\left\{x_t^2 - \frac{x_t^2 + x^2\Delta t^2 + \dot{2}\, x_t\dot{x}\Delta t}{x_t^2(x_t + \dot{x}\Delta t)}\right\}$$

$$\hat{\dot{A}} = -\frac{C}{\Delta t}\dot{x}\left\{\frac{\dot{x}\Delta t + 2x_t\Delta t}{x_t^2(x_t + \dot{x}\Delta t)^2}\right\}$$

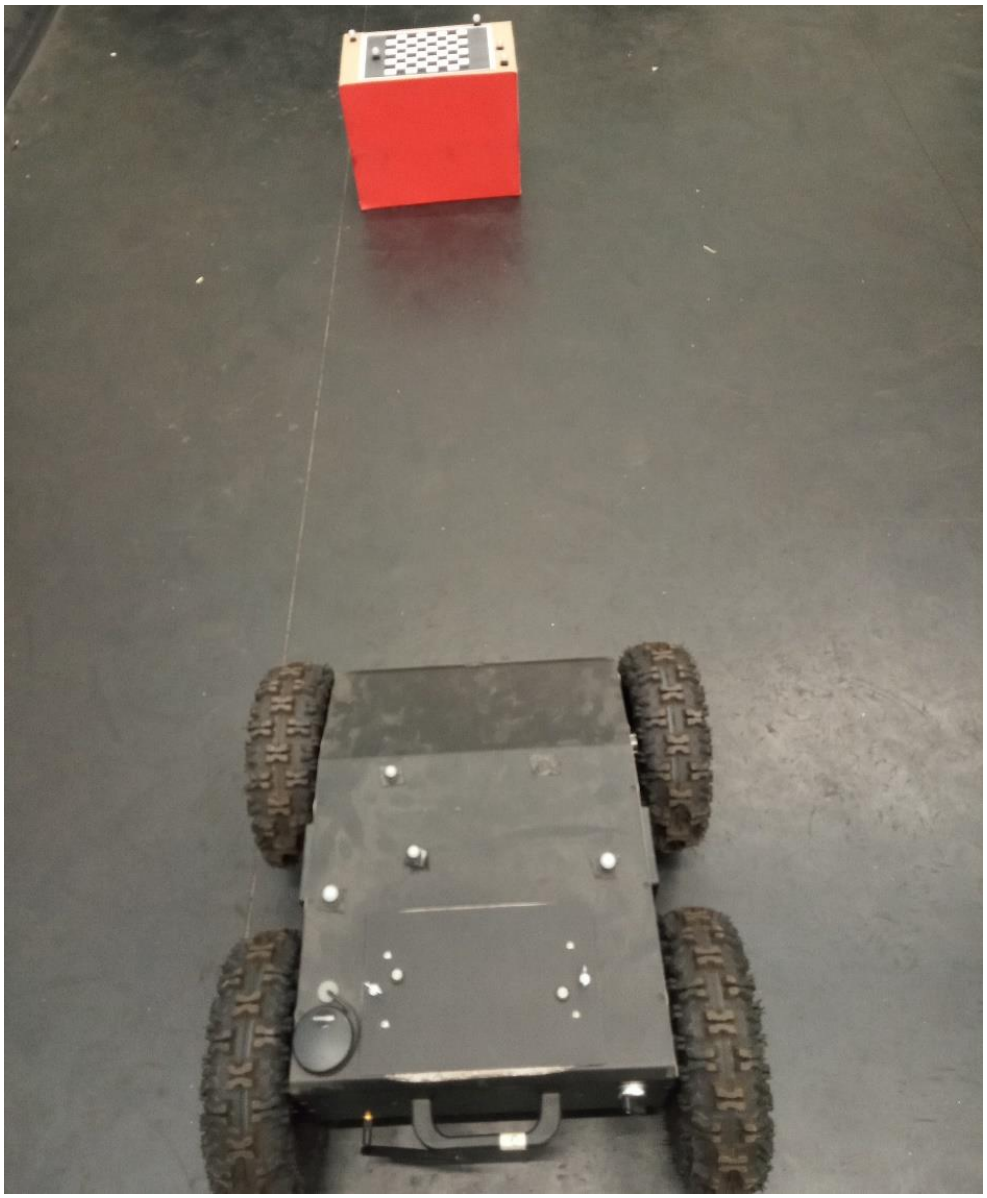
$$\hat{\tau} = \frac{2A}{\hat{\dot{A}}} = -\frac{2\frac{C}{x_t^2}}{\frac{C}{\Delta t}\,\dot{x}\left\{\frac{\dot{x}\Delta t^2 + 2x_t\Delta t}{x_t^2 + (x_t + \dot{x}\Delta t)^2}\right\}} = -2\,\frac{(x_t + \dot{x}\Delta t)^2\Delta t}{\dot{x}(\dot{x}t^2 + 2x_t\Delta t)}$$

$$\tau = \frac{2A}{\dot{A}} = \frac{2\frac{C}{x^2}}{-\frac{C}{x^3}\,\dot{x}} = 2\frac{x_t}{\dot{x}}$$

$$E = \tau - \hat{\tau} = 2\frac{x_t}{\dot{x}} - 2\frac{(x_t + \dot{x}\Delta t)^2 \Delta t}{\dot{x}(\dot{x}t^2 + 2x_t\Delta t)}$$

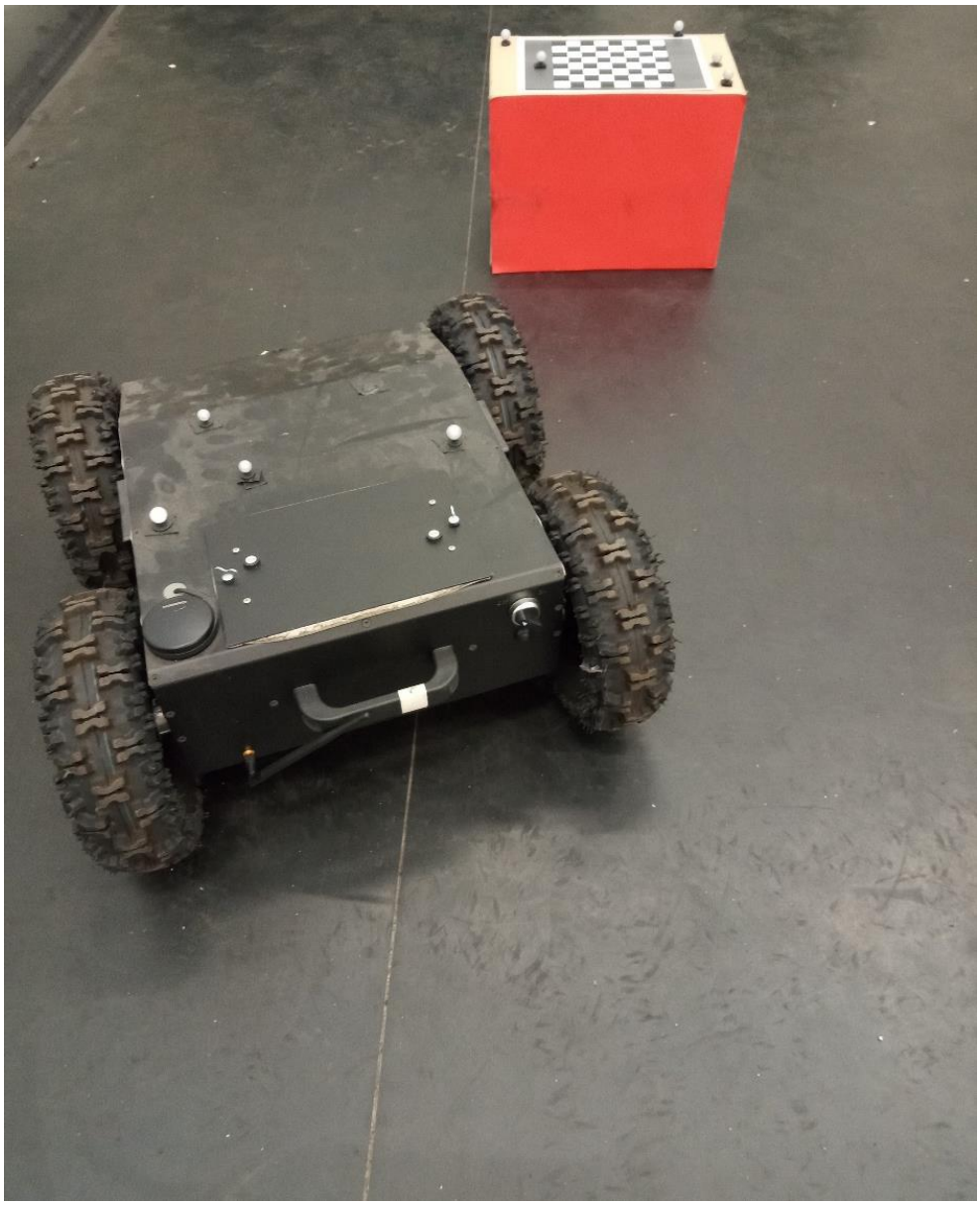
APPENDIX B

This appendix shows a series of images from the lateral manoeuvre experiment.

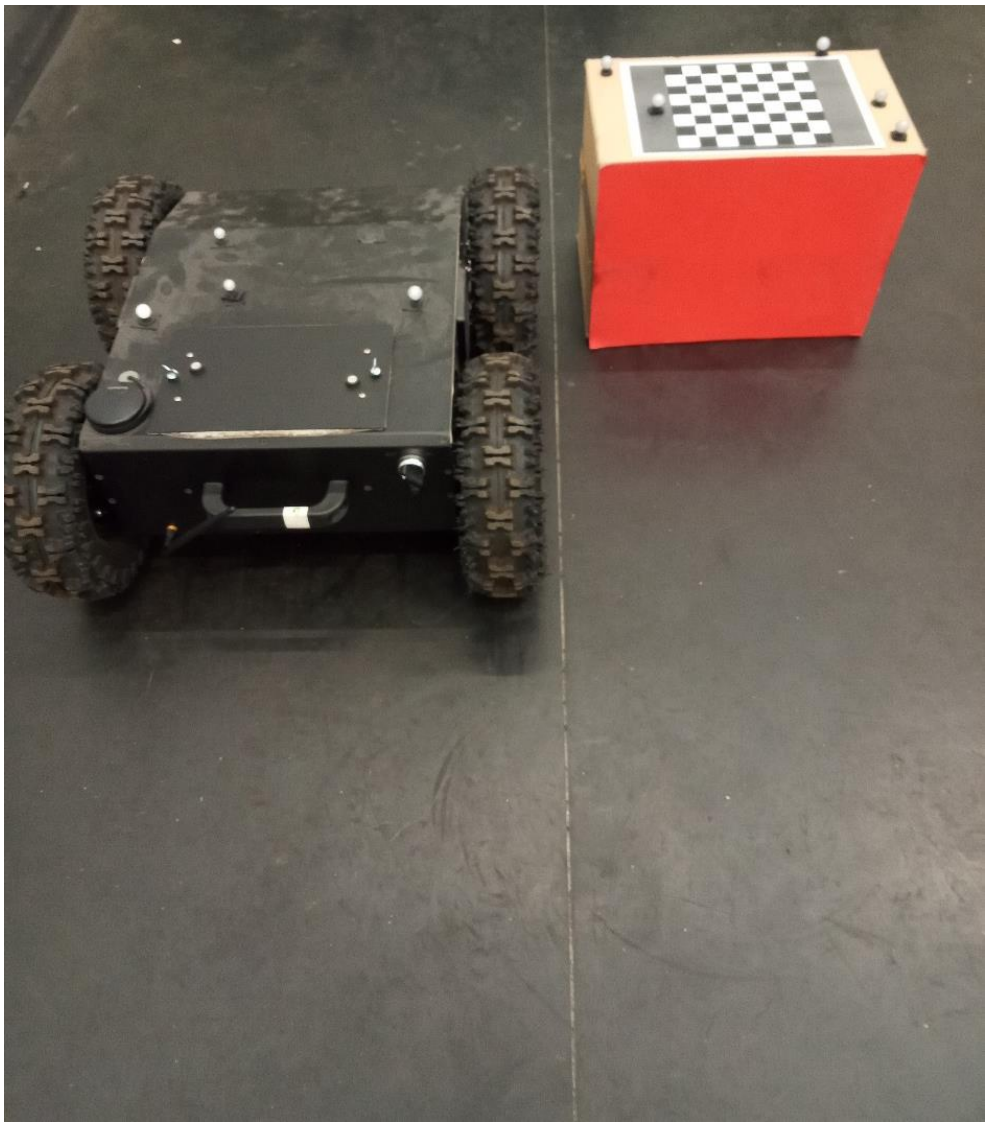


The above shows that there is an obstacle in the way of the rover.

The manoeuvre has begun, the robot has turned left to manoeuvre around the obstacle.



The rover is now turning right in order to return to its original orientation.



The manoeuvre is now complete, the obstacle has been avoided and the rover can now continue its original heading.